



# KiYOUNG2

P Stage 2: Open-Domain Question Answering Task Solution sharing

한국어 오픈 도메인 기계 독해 대회 1등 솔루션 공유

2021.10.11 ~ 2021.11.05

14조: 김대웅 김태욱 김채은 유영재 이하람 진명훈 허진규

# INDEX

---

- 01 Introduction
- 02 ODQA task
- 03 Strategy
- 04 Conclusion
- 05 KiYOUNG2

# 01. Introduction

팀 기영이에 대한 소개

대회의 목적

4주간의 계획 수립

Korean is all **YOU** Need for dialoGuE



K I Y O U N G



안녕하세요! 팀 기영이 입니다 😊

### 이번 대회의 목표

- 개인의 성장과 팀 협업을 우선 순위로!
- LB 보면서 스트레스 받지 않기!
  - 1주차 제출 금지 → baseline 완벽하게 이해하고 모듈화
- 코드 추상화
- 각자 아이디어를 맡아 구현하기
- 최대한 다양한 방법으로 학습하여 양상을 시너지 올리기
- ODQA task에 대한 이해력 증진

### Planning

- 1주차 : baseline 코드 분석 후 모듈화 / 2, 3주차 구현 Idea 제안
- 2주차 : Retriever, Reader 코드 추상화 및 2주차 아이디어 구현
- 3주차 : 각 모델 성능 향상을 위한 3주차 아이디어 구현 및 실험
- 4주차 : 최종 모델 튜닝 및 양상을 실시

- Retrieval: BM25, ElasticSearch, DPR
- Extractive + QA Conv Head
- Paraphrase generation
- Random Masking / KoEDA, ADEA, BT

- Wiki data augmentation
- Bart Denoising
- Curriculum Learning
- Query / Morpheme Analyzer Ensemble

## 02. ODQA task

Open-Domain Question Answering이란?

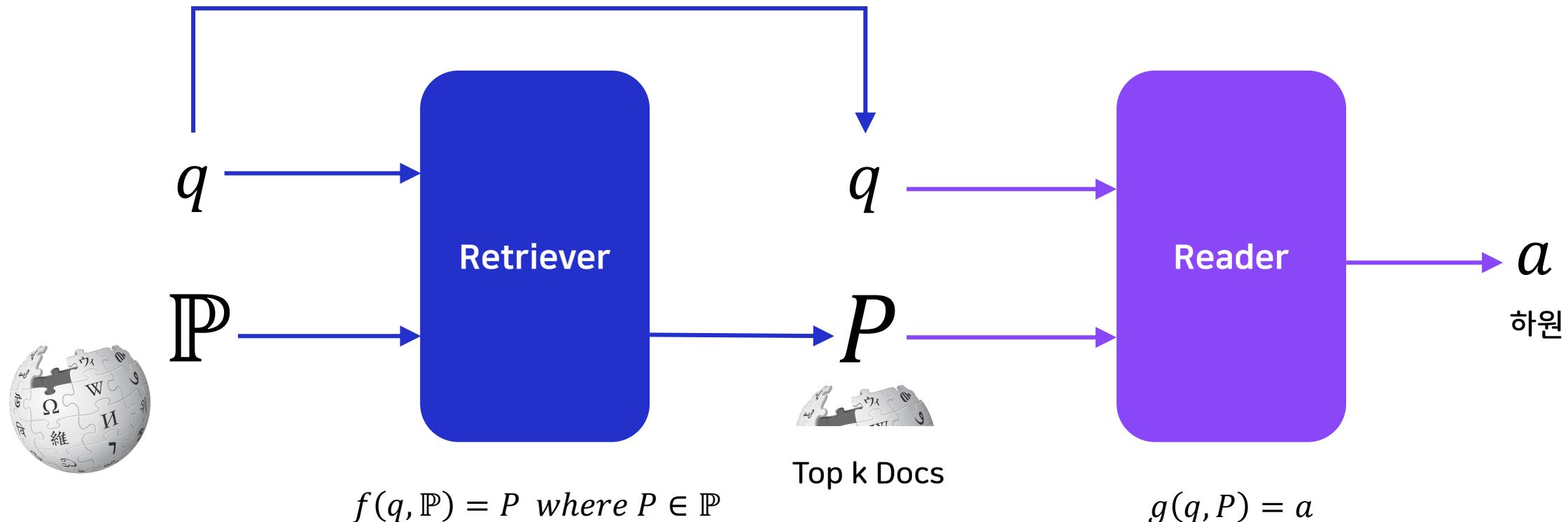
Dataset Details

Evaluation Metrics

어떻게 ODQA 문제를 풀었는지?

## What is Open-Domain Question Answering?

대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?



지문이 주어지지 않은 상태에서 방대한 World Knowledge에 기반하여 질의응답!

## Dataset Details



### 데이터 구성

분류(디렉토리 명)	세부 분류	샘플 수	용도	공개여부
train_dataset	train	3952	학습용	모든 정보 공개 (id, question, context, answers, document_id, title)
	validation	240		
test_dataset	validation	240 (Public)	제출용	id, question 만 공개
		360 (Private)		

약 57,000개의 unique한 Wikipedia documents도!

### 데이터 예시

#### context: 예측에 사용될 문맥 데이터

미국 상의원 또는 미국 상원(United States Senate)은 양원제의 미국 의회의 상원이다. 미국 대통령이 상원의 장이 된다. 각 주당 2명의 상원의원이 선출되어 100명의 상원의원으로 구성되어 있다. 임기는 6년이며, 2년마다 50개주 중 1/3씩 상원의원을 새로 선출하여 연방에 보낸다. 미국 상원은 미국 하원과는 다르게 미국 대통령을 수반으로 하는 미국 연방 행정부에 각종 동의를 하는 기관이다. 하원이 세금과 경제에 대한 권한, 대통령을 포함한 대다수의 공무원을 파면할 권한을 갖고 있는 국민을 대표하는 기관인 반면 상원은 미국의 주를 대표한다. 즉 캘리포니아주, 일리노이주 같이 주 정부와 주 의회를 대표하는 기관이다. 그로 인하여 군대의 파병, 관료의 임명에 대한 동의, 외국 조약에 대한 승인 등 신속을 요하는 권한은 모두 상원에게만 있다. 그리고 하원에 대한 견제 역할(하원의 법안을 거부할 권한 등)을 담당한다. 2년의 임기로 인하여 급진적인 수밖에 없는 하원은 지나치게 급진적인 법안을 만들기 쉽다. 대표적인 예로 건강보험 개혁 당시 하원이 미국 연방 행정부에게 퍼블릭 옵션(공공건강보험기관)의 조항이 있는 반면 상원의 경우 하원안이 지나치게 세금이 많이 든다는 이유로 퍼블릭 옵션 조항을 제외하고 비영리건강보험기관이나 보험회사가 담당하도록 한 것이다. 이 경우처럼 상원은 하원이나 내각책임제가 빠지기 쉬운 국가들의 국회처럼 걸핏하면 발생하는 의회의 비정상적인 사태를 방지하는 기관이다. 상원은 급박한 처리사항의 경우가 아니면 법안을 먼저 내는 경우가 드물고 하원이 만든 법안을 수정하여 다시 하원에 되돌려보낸다. 이러한 방식으로 단원제가 빠지기 쉬운 함정을 미리 방지하는 것이다. 날짜=2017-02-05

#### question: 예측에 사용될 질문 데이터

대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은? → 모든 query data는 의문문 형태

#### title: context의 주제

미국 상원 → Retrieval에 사용할 경우  
검색 확률이 올라감

#### id: sample의 고유 id

mrc-1-000067

#### document\_id: document의 고유 id

18293

#### answers: 정답의 위치와 text를 기록

{'answer\_start': [235], 'text': ['하원']}

## Exact Match

모델의 예측과 실제 답이 정확하게 일치하는 경우만 점수를 부여 (즉, 0 or 1)

- 띄어쓰기나 “.”과 같은 문자는 제거하고 평가
- 복수 정답의 경우 하나라도 일치하면 정답으로 간주

In 1870, Tesla moved to Karlovac, to attend school at the Higher Real Gymnasium, where he was profoundly influenced by a math teacher Martin Sekulić.:32 The classes were held in German, as it was a school within the Austro-Hungarian Military Frontier. Tesla was able to perform integral calculus in his head, which prompted his teachers to believe that he was cheating. He finished a four-year term in three years, graduating in 1873.:33

Why did Tesla go to Karlovac?

Ground Truth Answers: to attend school to attend school attend school at the Higher Real Gymnasium

Prediction: to attend school at the Higher Real Gymnasium

## F1 score

EM과 다르게 부분 점수를 제공

- Real: Barack Obama, Pred: Obama일 때 EM은 0점이지만 F1 score는 부분 점수를 부여
- 리더보드 순위로 사용하진 않고 참고용 지표

The definition of true positive (TP), true negative (TN), false positive (FP), false negative (FN)

	Tokens in Reference	Tokens Not in Reference
tokens in candidate	TP	FP
tokens not in candidate	FN	TN

$$Precision = \frac{\text{num}(\text{same\_token})}{\text{num}(\text{pred\_tokens})}$$

$$Recall = \frac{\text{num}(\text{same\_token})}{\text{num}(\text{groud\_tokens})}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

### ODQA 문제 풀이에 어떻게 접근할까?

- 기계 학습 → 인간의 학습 방식에서 아이디어를 착안
- Purpose: 사람과 같이 학습하는 모델을 구축하자!
- 사람과 같이? 사람은 어떻게 학습할까?
  - 우리는 중요할 것이라 생각되는 부분에 밑줄을 긋는다. ([Underlining](#))
  - 초-중-고의 순으로 국가에서 정해준 커리큘럼을 이수한다. ([Curriculum Learning](#))
  - 사람마다 학습을 위해 참고하는 자료가 다르다. ([Data Augmentation](#))
  - 질문에 대한 답변을 모델 스스로 생성한다. ([Generative Model](#))
- Validation Set에서 오답들을 분석한 결과?
  - [Reader 문제](#)] 날짜 문제를 잘 못 풀더라! → PORORO 모델의 기학습 가중치 활용
  - [Reader 문제](#)] 뒤에 조사가 붙은 채로 나오는 결과가 많더라! → 형태소 분석기 양상을 활용
  - [Reader 문제](#)] 복잡한 의미 관계 추론을 힘들어 하더라! → 다양한 데이터로 다양한 모델에 태워서 양상을
  - [Retrieval 문제](#)] 이상한 문서를 가져오더라! → Query 양상을 + Title을 Context에 붙이기
- 2주간의 baseline 모듈화 과정 중 Reader가 잘 못 읽는 경우가 많다는 것을 캐치
- 사람같이 읽고 집단지성으로 더 우수해지는 것과 같이 양상을 시너지가 나도록 다양하게 학습시키는 것에 방점!

# 03. Strategy

Retrieval / Process wiki data

MRC Data Augmentation

Enhanced Model

Curriculum Learning

Query / Morpheme Analyzer Ensemble

## 가장 잘 찾아오는 Retrieval Setting 찾기!

```

from solution.retrieval import RETRIEVAL_HOST

RETRIEVAL_HOST["elastic_engine"]["elastic_search"]

solution.retrieval.elastic_engine.api.ESRetrieval

data_args.rebuilt_index = False

retriever = RETRIEVAL_HOST["elastic_engine"]["elastic_search"](data_args)

Lengths of unique contexts : 56737
    
```

Aa	# bm25적용...	# dfr 적용 후	# bm25성능개...	# dfr성능개선...	▼ 우위 모델
top-5	86.71278626	86.85591603	0.333969466	0.143129771	DFR
top-10	90.74427481	90.43416031	1.049618321	0.166984733	BM25
top-15	92.67652672	92.1278626	1.073473282	0.143129771	BM25
top-20	93.70229008	93.39217557	1.121183206	0.190839695	BM25
top-25	94.15553435	94.17938931	0.834923664	0.214694656	DFR
top-30	94.65648855	94.65648855	0.78721374	0.238549618	DFR
top-35	94.94274809	95.0620229	0.691793893	0.381679389	DFR

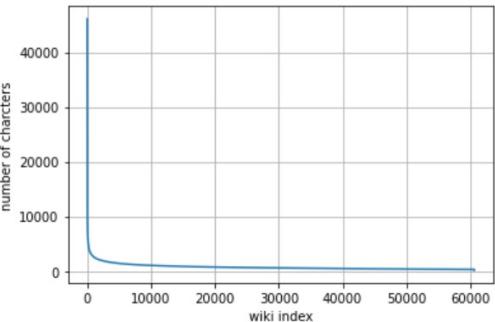
```

'settings':{
  'analysis':{
    'analyzer':{
      'my_analyzer':{
        'type': "custom",
        'tokenizer':'nori_tokenizer',
        'decompound_mode':'mixed',
        'stopwords': '_korean_',
        'synonyms': '_korean_',
        # filtering
        "filter": [
          "lowercase", #영문 소문자로 변경
          "my_stop_filter", #사용자가 설정한 filter
          "nori_readingform", #한자 음독 변환
          "cjk_bigram", #chinese-japaness-korean
          "decimal_digit" #아라비아 숫자 외에 문자를 전부 아라비아 숫자로 변경
        ],
      },
    },
    'filter':{
      "my_stop_filter": {
        "type" : "stop",
        "stopwords_path" : "user_dic/my_stop_dic.txt"
      }
    }
  },
  'similarity':{
    # setting ranking functions
    'my_bm25_similarity':{
      'type': 'BM25',
      'b': 0.75, #[0.3 ~ 0.8]
      'k1': 1.2 #[1.2 ~ 2.0]
    },
    'my_dfr_similarity':{
      'type': 'DFR',
      "basic_model": "g", #[g, if, in, ine]
      "after_effect": "l", #[b, l]
      "normalization": "h2", #[no, h1, h2, h3, z]
      "normalization.h2.c": 3.0 #[1.0 ~ 3.0]
    },
    'my_dfi_similarity':{
      'type': 'DFI',
      "independence_measure": "standardized", #[standardized, saturated, chisquared]
    },
    'my_ib_similarity':{
      'type': 'IB',
      "distribution": "ll", #[ll, spl]
      "lambda": "df", #[df, ttf]
      "normalization": "h2", #[no, h1, h2, h3, z]
    },
    # LM Dirichlet
    'my_lmd_similarity':{
      'type': 'LMDirichlet',
      "mu": 2000 #단어 빈도 관련 페널티
    },
    #LM Jelinek Mercer
    'my_lmjm_similarity':{
      'type': 'LMJelinekMercer',
      "lambda": 0.1 #[0.1(short text) ~ 0.7(long text)]
    }
  }
}
    
```

전부 arguments로  
control 가능하게 코딩!

## Wiki documents 데이터 처리!

- Title을 Context에 붙이면 검색 성능이 올라감!
- Wiki 60,613(중복 포함) 문서에 대해 아래의 처리를 수행
  - 간단한 전처리 수행
  - 중복 제거 (baseline 코드로 수행하는 것과 전처리를 하고 제거하는 것에 차이가 있음)
  - Char length 3000>=인 문서 직접 필터링 (약 3,000개 직접 눈으로 보고 제거)
  - 한글 비율이 70% 밑이면 제거
  - “현장 한역”이 들어있는 중국 문서 제거
  - Char length 1000>인 문서 char 단위로 분리
  - Kss를 활용하여 문장으로 분리
    - 성경 구절 같이 kss가 못 자르는 문서 1,163개 대해 직접 확인 후 자름
  - 최대 길이 800을 넘지 않게 문장 단위로 문서를 합쳐줌
  - 위 작업으로 약 7,000개의 문서를 제거했고 결과적으로 56,737개의 문서를 생성
  - Elasticsearch의 bulk api를 활용하여 build 및 indexing 시간을 1/4로 줄임



Char의 수가 비정상적으로 많은 context  
들을 숙아낼 필요가 있음!

```

def build_index(self, index_name: str):
    assert not self.is_exists_index()
    t0 = time.time()
    print(f"Create elasticsearch index: {index_name}")
    index_config = self.build_index_config()
    self.engine.indices.create(index=index_name, body=index_config, ignore=400)
    document_texts = [
        {"_id": i,
         "_index": self.index_name,
         "_source": {"document_text": doc}}
        for i, doc in enumerate(self.contexts)]
    helpers.bulk(self.engine, document_texts)
    with open("configs/es_index_config.json", "w", encoding="utf-8") as f:
        json.dump(index_config, f)
    print(f"Done {time.time() - t0:.3f}")

def make_query(self, query, topk):
    return {"query": {"match": {"document_text": query}}, "size": topk}

def get_relevant_doc(self, query_or_dataset, topk):
    if isinstance(query_or_dataset, Dataset):
        query = query_or_dataset["question"]
    elif isinstance(query_or_dataset, str):
        query = [query_or_dataset]
    elif isinstance(query_or_dataset, list):
        query = query_or_dataset
    else:
        raise NotImplementedError
    body = []
    for i in range(len(query)*2):
        if i % 2 == 0:
            body.append({"index": self.index_name})
        else:
            body.append(self.make_query(query[i//2], topk))

    response = self.engine.msearch(body=body)["responses"]

    doc_scores = [[hit["_score"] for hit in res["hits"]["hits"]] for res in response]
    doc_indices = [[hit["_id"] for hit in res["hits"]["hits"]] for res in response]
    doc_contexts = [[hit["_source"]["document_text"] for hit in res["hits"]["hits"]] for res in response]

    return doc_scores, doc_indices, doc_contexts

```

Bulk api로 속도 향상!

## 다양한 데이터로 잘 읽는 모델 만들기!

- By [TUNiB](#), 다양한 모델, 다양한 데이터로 학습한 결과를 양상볼하면 시너지가 좋음!
- 우리의 직관과도 일치함
  - 인간은 다른 학원을 다니고 다른 학습지 혹은 과외 선생님한테 insight를 얻음
  - 무지개 빛깔만큼 다양한 사람들이 모이면 더 큰 시너지를 보일 것!
- 위 원칙에 따라 대회 기간 동안 아래 작업에 착수함
  - PORORO MRC + Question Generation (<https://github.com/kakaobrain/pororo>)
  - Paraphrasing Generation (<https://github.com/kakaobrain/pororo>)
  - Pivot Translation (<https://github.com/jinmang2/PororoBT>)
  - Random Masking (독자적인 코드)
  - Sentence Permutation ([https://github.com/pytorch/fairseq/blob/main/fairseq/data/denoising\\_dataset.py#L218](https://github.com/pytorch/fairseq/blob/main/fairseq/data/denoising_dataset.py#L218))
  - KoEDA, ADEA Augmentation (<https://github.com/toriving/KoEDA>)
  - Add context summary (<https://kakaobrain.github.io/pororo/seq2seq/summary.html>)
- 위 데이터를 Versioning하고 curriculum learning을 통해 극적인 성능 향상을 도모했음 (양상볼에서!)

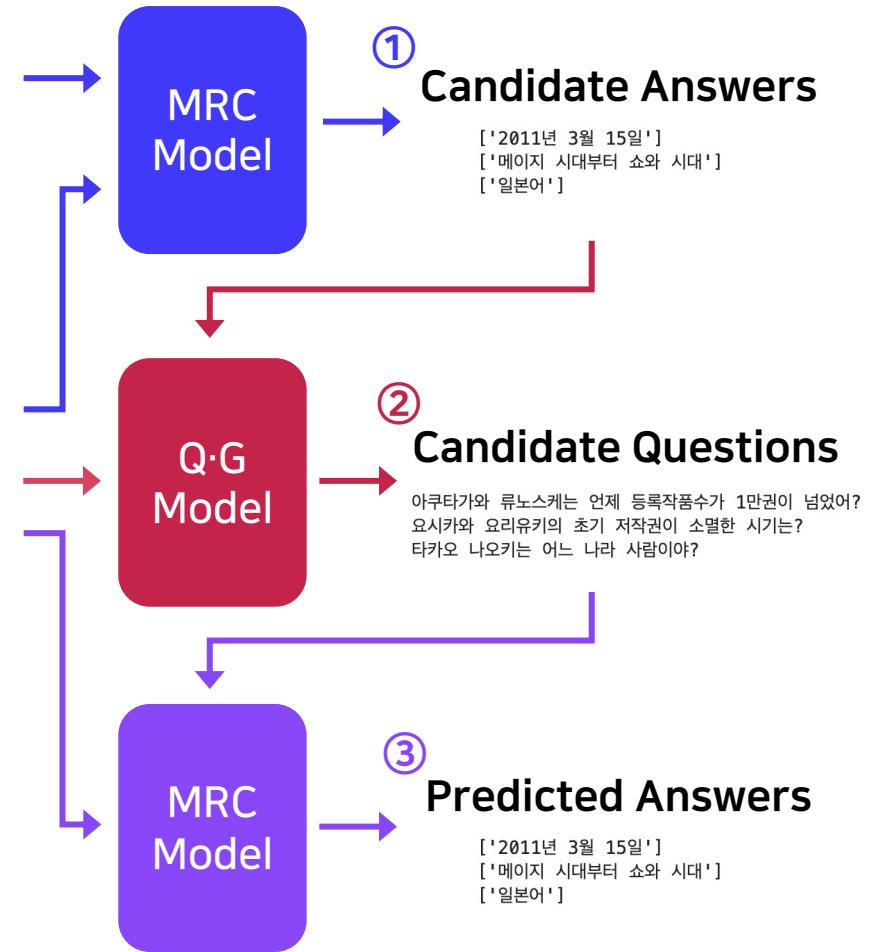
# 다양한 데이터로 잘 읽는 모델 만들기!

## Based question(45개)

Who : "누구야?", "사람은?", "선수는?", "감독은?",  
 When : "언제야?", "시기는?", "날짜는?", "사망일은?", "연도는?", "때는?", "며칠인가?"  
 Where : "어디야?", "거주지는?"  
 What : "뭐야?", "이름은?", "회사는?", "나라는?", "상대는?", "소속은?", "기관은?", "무슨 책인가?", "무슨 작품인가?",  
 "기준은?", "제목은?", "명칭은?", "칭호는?", "용어는?", "사건은?", "종류는?", "도구는?", "동물은?"  
 How : "얼마야?", "방법은?", "개수는?", "몇 개인가?", "금액은?", "온도는?", "높이는?", "길이는?", "인구는?"  
 Why : "원인은?", "근거는?", "원리는?", "계기는?", "끼닭은?"

## Context (wiki\_documents, train\_datasets, valid\_datasets)

미국 상의원 또는 미국 상원(United States Senate)은 양원제인 미국 의회의 상원이다. 미국 부통령이 상원의장이 된다. 각 주당 2명의 상원의원이 선출되어 100명의 상원의원으로 구성되어 있다. 임기는 6년이며, 2년마다 50개주 중 1/3씩 상원의원을 새로 선출하여 연방에 보낸다. 미국 상원은 미국 하원과는 다르게 미국 대통령을 수반으로 하는 미국 연방 행정부에 각종 동의를 하는 기관이다. 하원이 세금과 경제에 대한 권한, 대통령을 포함한 대다수의 공무원을 파면할 권한을 갖고 있는 국민을 대표하는 기관인 반면 상원은 미국의 주를 대표한다. 즉 캘리포니아주, 일리노이주 같이 주 정부와 주 의회를 대표하는 기관이다. 그로 인하여 군대의 파병, 관료의 임명에 대한 동의, 외국 조약에 대한 승인 등 신속을 요하는 권한은 모두 상원에게만 있다. 그리고 하원에 대한 견제 역할(하원의 법안을 거부할 권한 등)을 담당한다. 2년의 임기로 인하여 급진적일 수밖에 없는 하원은 지나치게 급진적인 법안을 만들기 쉽다. 대표적인 예로 건강보험 개혁 당시 하원이 미국 연방 행정부에게 퍼블릭 옵션(공공건강보험기관)의 조항이 있는 반면 상원의 경우 하원이 지나치게 세금이 많이 듦다는 이유로 퍼블릭 옵션 조항을 제외하고 비영리건강보험기관이나 보험회사가 담당하도록 한 것이다. 이 경우처럼 상원은 하원이나 내각책임제가 빠지기 쉬운 국가들의 국회처럼 걸핏하면 발생하는 의회의 비정상적인 사태를 방지하는 기관이다. 상원은 급박한 처리사항의 경우가 아니면 법안을 먼저 내는 경우가 드물고 하원이 만든 법안을 수정하여 다시 하원에 되돌려보낸다. 이러한 방식으로 단원제가 빠지기 쉬운 함정을 미리 방지하는 것이다. 날짜=2017-02-05



Context에 대해 다양한 Query를 생성하자!



## Context를 비틀면 모델이 더 robust하게 학습하지 않을까?

→ Masking

Question과 유사한 정답이 아닌 token을 나이도에 따라 교정

	Easy	해당 Token을 [MASK]로 교체
	Normal	원본 Context 유지
	Hard	해당 Token을 반복하여 추가

## Context

미국 상의원 또는 미국 상원(United States Senate)은 양원제인 미국 의회의 상원이다. 미국 대통령이 상원의장이 된다. 각 주당 2명의 상원의원이 선출되어 100명의 상원의원으로 구성되어 있다. 임기는 6년이며, 2년마다 50개주 중 1/3씩 상원의원을 새로 선출하여 연방에 보낸다. 미국 상원은 미국 하원과는 다르게 미국 대통령을 수반으로 하는 미국 연방 행정부에 각종 동의를 하는 기관이다. 하원이 세금과 경제에 대한 권한, 대통령을 포함한 대다수의 공무원을 파면할 권한을 갖고 있는 국민을 대표하는 기관인 반면 상원은 미국의 주를 대표한다. 즉 캘리포니아주, 일리노이주 같이 주 정부와 주 의회를 대표하는 기관이다. 그로 인하여 군대의 파병, 관료의 임명에 대한 동의, 외국 조약에 대한 승인 등 신속을 요하는 권한은 모두 상원에게만 있다. 그리고 하원에 대한 견제 역할(하원의 법안을 거부할 권한 등)을 담당한다. 2년의 임기로 인하여 급진적일 수밖에 없는 하원은 지나치게 급진적인 법안을 만들기 쉽다. 대표적인 예로 건강보험 개혁 당시 하원이 미국 연방 행정부에게 퍼블릭 옵션(공공건강보험기관)의 조항이 있는 반면 상원의 경우 하원안이 지나치게 세금이 많이 든다는 이유로 퍼블릭 옵션 조항을 제외하고 비영리건강보험기관이나 보험회사가 담당하도록 한 것이다. 이 경우처럼 상원은 하원이나 →내각책임제가 빠지기 쉬운 국가들의 국회처럼 걸핏하면 발생하는 의회의 비정상적인 사태를 방지하는 기관이다. 상원은 급박한 처리사항의 경우가 아니면 법안을 먼저 내는 경우가 드물고 하원이 만든 법안을 수정하여 다시 하원에 되돌려보낸다. 이러한 방식으로 단원제가 빠지기 쉬운 함정을 미리 방지하는 것이다. 날짜=2017-02-05

## Question

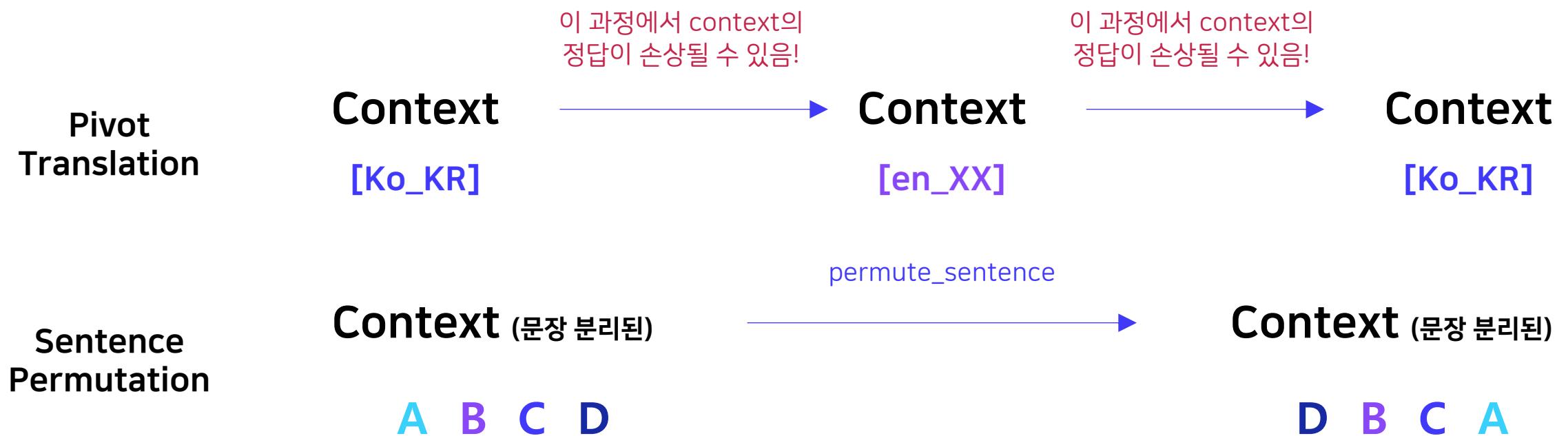
대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?

Tokens	이순신	##은	조선	중기	무신	##이다.
Word-level High Score Masking	이순신	##은	조선	중기	[MASK]	
basic	이순신	##은	조선	중기	무신	##이다.
Word-level High Score Adding	이순신	##은	조선	중기	무신	무신무신이다.

## Context를 비틀면 모델이 더 robust하게 학습하지 않을까?

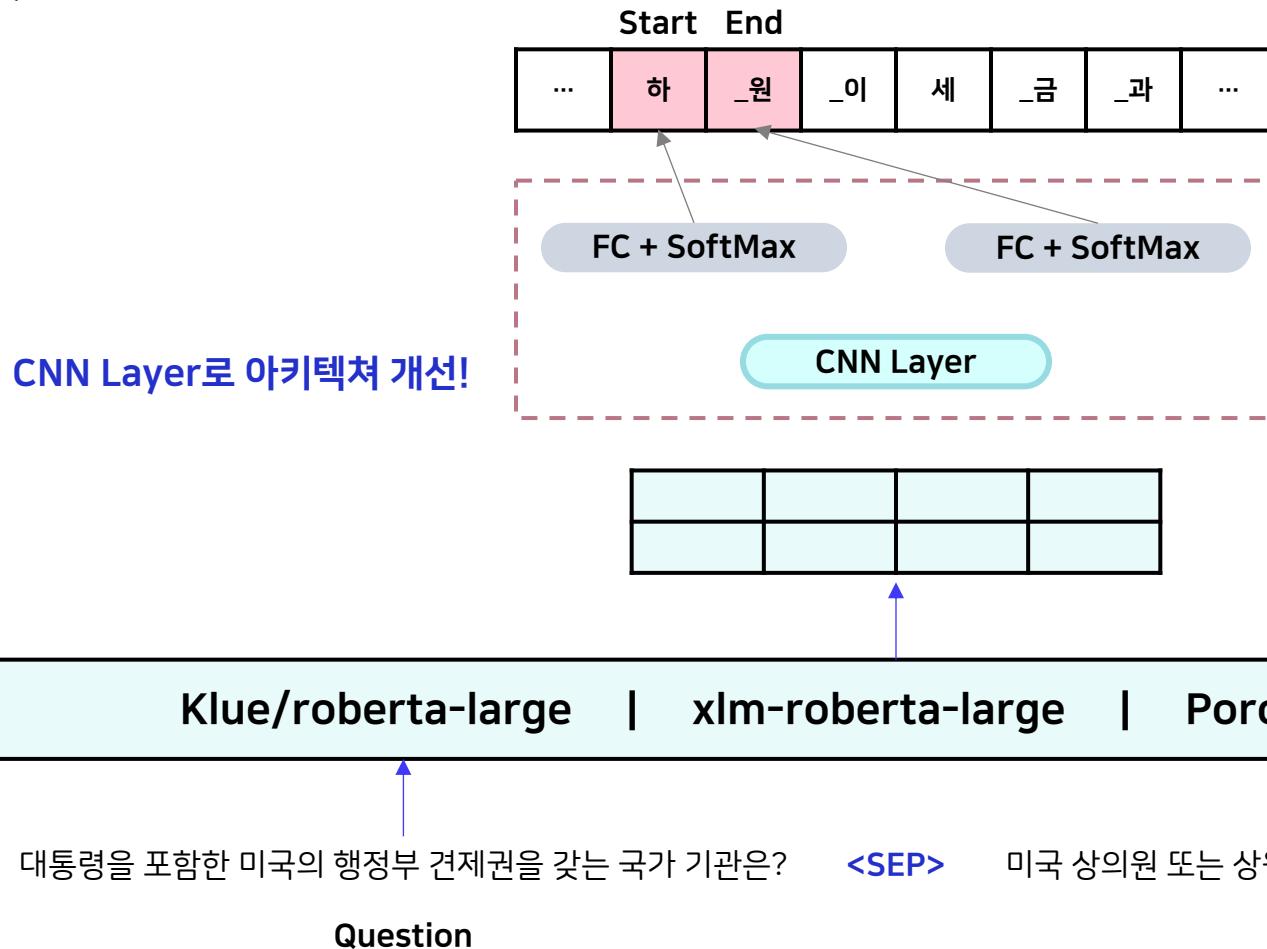
→ Pivot Translation / Sentence Permutation

- PT) 한국어 -> 영어 -> 한국어
- PT) Context 내에 정답이 없거나 2개 이상인 데이터 삭제, 정답 시작 위치 재설정
- SP) 문서를 KSS로 문장 단위로 분리한 다음 임의로 permutation



## 활용 모델 및 일반적인 MRC 학습 과정

→ 두 개의 최종 Layer를 통해서 각각 정답의 **시작과 끝을 예측**



```
class QAConvSDSLayer(nn.Module):
    def __init__(self, input_size: int, hidden_dim: int):
        super().__init__()
        self.conv1 = nn.Conv1d(
            in_channels=input_size,
            out_channels=input_size*2,
            kernel_size=3,
            padding=1
        )
        self.conv2 = nn.Conv1d(
            in_channels=input_size*2,
            out_channels=input_size,
            kernel_size=1,
        )
        self.layer_norm = nn.LayerNorm(hidden_dim)

    def forward(self, x):
        out = self.conv1(x)
        out = self.conv2(out)
        out = x + torch.relu(out)
        out = self.layer_norm(out)
        return out
```

## CNN을 활용한 아키텍처 개선

Samsung SDS의 구현을 reproduce!

Input shape을 유지하면서 근접 벡터 간 연관 정보를 학습하도록 설계됨

CNN Layer는 5개 층, 1D Conv + ReLU, LayerNorm, Residual Connection 적용

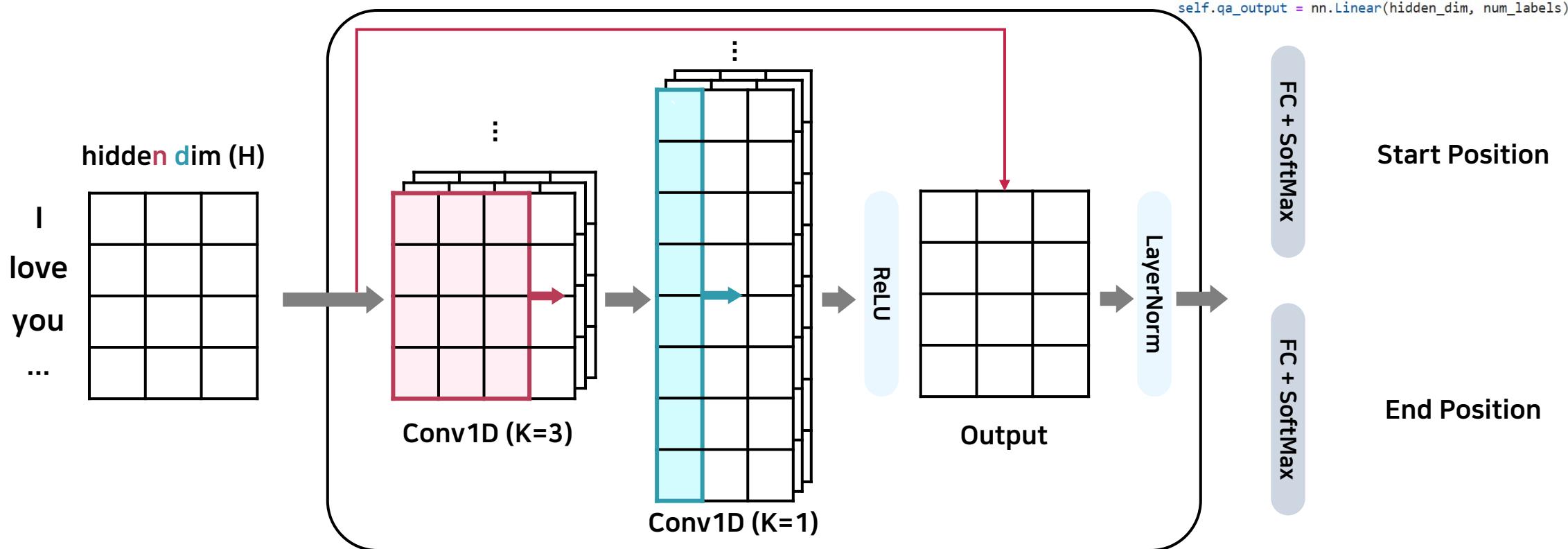
Kernel0| hidden dim 방향으로 움직임!

```
import torch
import torch.nn as nn
from solution.reader.architectures import QAConvSDSHead

sds_head = QAConvSDSHead(512, 768, 5, 2)
encoder_outputs = torch.randn(32, 512, 768) # bsz, seq_len, hid_dim
sds_head.convs(encoder_outputs).shape # same shape!

torch.Size([32, 512, 768])    convs = []
for n in range(n_layers):
    convs.append(QAConvSDSLayer(input_size, hidden_dim))
self.convs = nn.Sequential(*convs)
self.qa_output = nn.Linear(hidden_dim, num_labels)
```

CNN Layer X 5



“ 핵심 문장을 강조해서 읽도록 punctuation 및 underline embedding layer 추가 ”

어? 그러면 추론 때는 어떻게 하나요?

→ 문장 별 유사도를 구한 모델로 정답이 있을 확률이 가장 높은 5개 문장에 밑줄을 그어줍니다!

### train dataset

- 정답이 포함된 문장 양 끝에 punctuation추가
- 정답이 포함된 문장은 1로 embedding

### test dataset

- 질문과 유사도가 높은 문장에 punctuation추가
- 질문과 유사도가 높은 문장은 1로 embedding

### 예시

- question : MRC대회 기간은?
- answer : 4주
- context : 어느덧 11월이다. ^4주간의 긴 MRC대회가 끝났다. ※ 다음은 최적화 대회다.

0000000000111111111111111100000000000000

punctuation

underline embedding

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	# #ing	[SEP]
Token Embeddings	E <sub>[CLS]</sub>	E <sub>my</sub>	E <sub>dog</sub>	E <sub>is</sub>	E <sub>cute</sub>	E <sub>[SEP]</sub>	E <sub>he</sub>	E <sub>likes</sub>	E <sub>play</sub>	E <sub>#ing</sub>	E <sub>[SEP]</sub>
Segment Embeddings	+ E <sub>A</sub>	+ E <sub>B</sub>	+ E <sub>B</sub>	+ E <sub>B</sub>	+ E <sub>B</sub>	+ E <sub>B</sub>					
Position Embeddings	E <sub>0</sub>	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	E <sub>8</sub>	E <sub>9</sub>	E <sub>10</sub>
underline Embeddings	+ E	+ E	+ E	+ E	+ E	+ E	+ E <sub>ans</sub>	+ E <sub>ans</sub>	+ E <sub>ans</sub>	+ E <sub>ans</sub>	+ E

## Curriculum Learning

### 어떤 난이도로 학습하면 효과적일까?

- 3가지 방식으로 난이도 별 데이터셋 구성 후 해당 시점에서 BEST 모델 3개로 각 샘플을 예측
- 샘플 별 모델이 맞춘 개수  $s_i$ 에 따라 level 0 ~ level 2의 3가지 난이도로 재구성
- 다음 두 가지 방식을 적용
  - Level 0 ~ 2까지 순차적으로 학습
  - Level 별 별도의 모델 학습 후 Ensemble

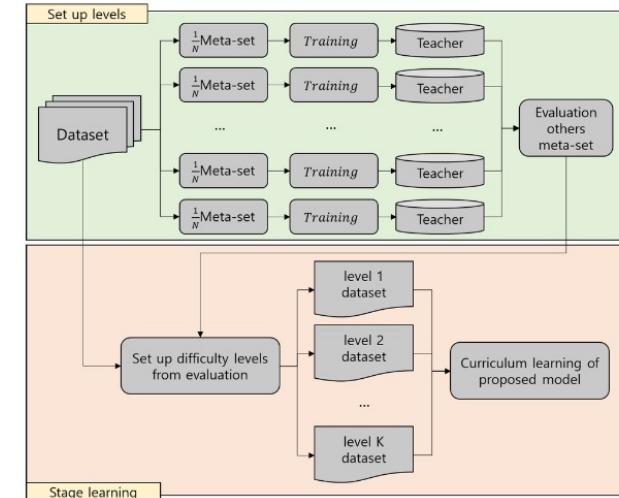
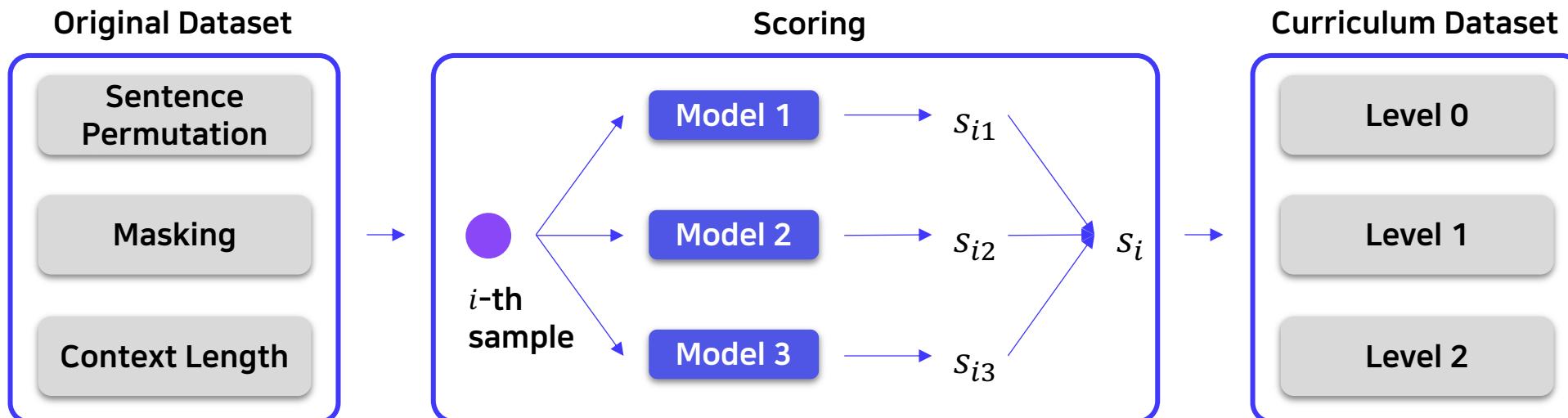


Figure 2: Curriculum learning process.  $K$  denotes the number of difficulty levels.



Okt, Mecab, Kkma, Komoran, Khaiii 사용!

## 형태소 분석기 양상블로 조사를 제거하여 EM 성능을 끌어올리자!

- 대부분의 정답이 '조사'를 빼어낸 형태
- 실제로 후처리 처리 여부에 따라 EM에서 4~5%의 차이를 보임
- 따라서 조사를 빼어내는 후처리 작업 진행!

### 질문 예시



지문

미국 상의원 또는 미국 상원(United States Senate)은 양원제인 미국 의회의 상원이다. 미국 부통령이 상원의장이 된다. 각 주당 2명의 상원의원이 선출되어 100명의 상원의원으로 구성되어 있다. 임기는 6년이며, 2년마다 50개주 중 1/3씩 상원의원을 새로 선출하여 연방에 보낸다. 미국 상원은 미국 하원과는 다르게 미국 대통령을 수반으로 하는 미국 연방 행정부에 각종 동의를 하는 기관이다.

하원이 세금과 경제에 대한 권한, 대통령을 포함한 대다수의 공무원을 파면할 권한을 갖고 있는 국민을 대표하는 기관인 반면 상원은 미국의 주를 대표한다  
..후략..

질문: 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?

정답: 하원

예측: 하원이

### 단어 단위 후처리 작업

하원이

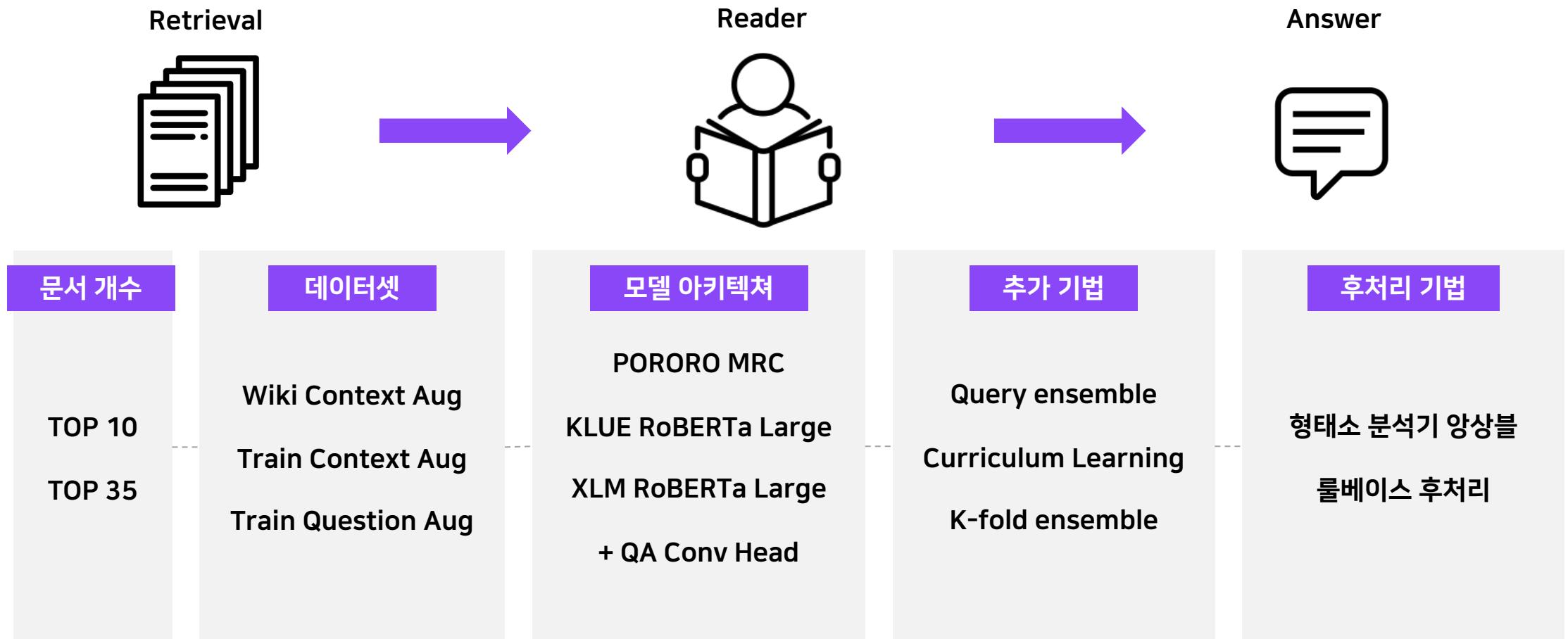
단어 단위로 수행할 경우 등록된 단어가 아니면 조사 제거에 실패하는 경우가 많음. 문장 단위로 보면 조사 파악에 용이!

### 문장 단위로 형태소 분석

미국 연방 행정부에 각종 동의를 하는 기관이다. 하원이 세금과 경제에 대한 권한, 대통령을 포함한

## 04. Conclusion

Final Pipeline  
LB results



최종 14개의 모델을 Hard-Voting하여 결과를 제출!

## LB results

Public

순위	팀 이름	팀 멤버	EM	F1	제출 횟수	최종 제출
1 (-)	MRC_14조	Ki YOUNG	76.250	86.460	64	18h
1	MRC_14조	Ki YOUNG	76.250	86.460	64	18h
2	MRC_9조	동건 꽃 보다	75.830	85.070	89	16h
3	MRC_2조	多样性图标	74.580	83.100	132	16h
4	MRC_5조	이그루	72.920	81.030	91	15h

Private

순위	팀 이름	팀 멤버	EM	F1	제출 횟수	최종 제출
1 (-)	MRC_14조	Ki YOUNG	75.560	84.250	64	15h
1	MRC_14조	Ki YOUNG	75.560	84.250	64	15h
2	MRC_9조	동건 꽃 보다	73.610	83.050	89	16h
3	MRC_8조	多样性图标	71.670	81.400	175	15h
4	MRC_1조	多样成员图标	70.280	80.180	45	15h

# 05. Appendix

Make code more readable with abstraction

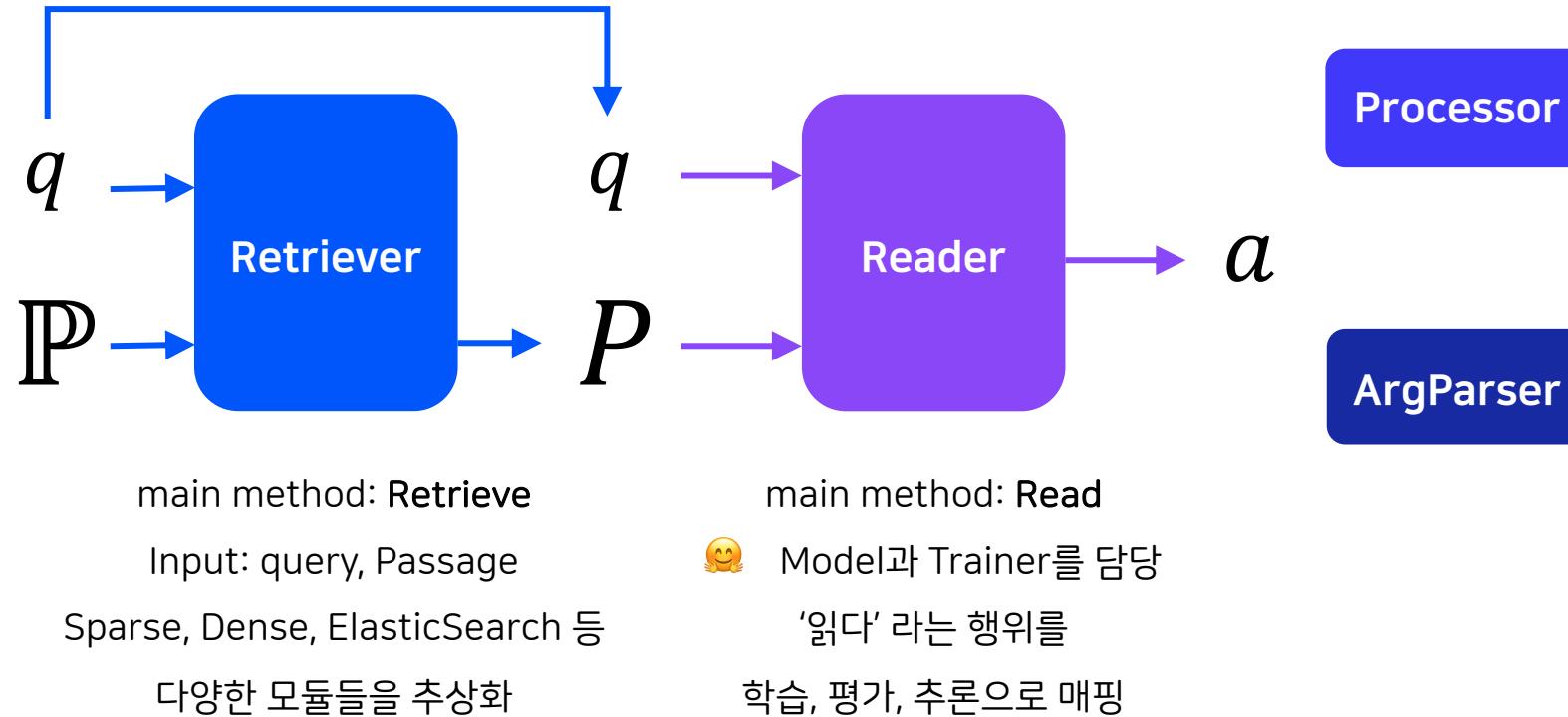
KANBAN Board

Dataset / Model Versioning

Team Seminar



## Make code more readable with abstraction



코드 추상화로 가독성과 빠른 개발 수행!


일정 By Status ▾

No Status 5

Sparse + Dense Retrieval Ensemble

Bi-RNN

진규 허 uyzae

양상블

November 2, 2021

DaeUng Kim

Retrospective Reader

October 31, 2021

Myunghoon Jin

[변경] SG-NET→ 모델 성능 개선을 위한 분석

November 1, 2021

DaeUng Kim 이하람

+ New

In Progress 4

Punctuation - 밑줄 짹악~!

November 2, 2021

진규 허 Myunghoon Jin uyzae  
 taeuk kim 님 채채

후처리(with. 형태소 분석기 양상블)

November 2, 2021

taeuk kim Myunghoon Jin  
 uyzae

Curriculum Learning

November 2, 2021

High   
 taeuk kim 진규 허  
 DaeUng Kim

Analyzer & Attention Visualization

October 24, 2021

High   
 DaeUng Kim Myunghoon Jin  
 이하람

+ New

Completed 7

(Noise) Bart의 Denoising Objectives

October 24, 2021

Myunghoon Jin taeuk kim

Question에 대해 KoEDA, AEDA, Backtranslation 적용

October 23, 2021

DaeUng Kim 님 채채 진규 허

코드 리팩토링

October 22, 2021

Myunghoon Jin

EDA 정리

taeuk kim 진규 허

ElasticSearch

Myunghoon Jin taeuk kim

uyzae

Paraphrasing Generation

October 23, 2021

DaeUng Kim uyzae

[Data] Human Labeling

October 24, 2021

High   
 DaeUng Kim Myunghoon Jin  
 uyzae 님 채채 진규 허  
 taeuk kim 이하람

+ New

Completed 2 7

DPR + Poly Encoder

November 2, 2021

Medium  
 Myunghoon Jin DaeUng Kim  
 이하람

Extraction based + Generation based MRC 양상블

이하람 님 채채

[Single] Generative Model

October 22, 2021

님 채채 이하람

(noising) Random Masking

October 22, 2021

진규 허 Myunghoon Jin

Context에 요약문 추가

October 24, 2021

Medium

DaeUng Kim 님 채채 이하람

N-gram Convolution layer 깊게 쌓기

October 22, 2021

taeuk kim Myunghoon Jin

Retrieval Experiment

October 22, 2021

Myunghoon Jin taeuk kim  
 uyzae

## KANBAN 보드로 각 작업 진행 상황 수시로 check

## Dataset / Model Versioning

Datasets: kiyoung2/aistage-mrc

like 4

Dataset card Files and versions Settings

### Version Info

- v4.1.1
- v4.1.0
- v4.0.1
- v4.0.0
- v3.2.3
- v3.2.2
- v3.2.1
- v3.2.0
- v3.1.0
- v3.0.0
- v2.1.1
- v2.1.0
- v2.0.1
- v2.0.0

### Dataset Preview

The dataset preview is not available for this dataset.

Go to dataset viewer

### AI Stage MRC task

### Version Info

#### v4.1.1

- v3.2.3 데이터셋 (train\_dataset\_aug)에 punctuation 추가한 데  
이터셋, both train and validation
- train\_aug\_punctuation에 있음

kiyoung2/roberta-large-qaconv-sds-aug

PyTorch Transformers roberta

Model card Files and versions Settings

### main

- LB 61.67, v1.3.1 train\_dataset\_aug 셋으로 학습

### rmjosa

- 조사 제거

### rmjosa\_bt

- 조사 제거 + 일본어 Back Translation

### wiki-pororoaug-only

- v1.3.0의 pororo\_aug(50,531건)으로만 학습한 모델

### tapt-wikipororo

- LB 62.5([TITLE], #제거 안함)
- wiki-pororoaug-only 모델에서 v1.3.1 train\_dataset\_aug 셋으로 학습



Huggingface Hub를 활용하여 데이터셋/모델 버저닝 → 빠른 실험과 공유!



@K	ElasticSearch 지식 공유	판타~스틱! 엘라~스틱!
@N	Generative Model	아 이름짓기 어렵다. Generative야 지어줘~
@N	Underline Module	핵심 문장을 파악하라!
@K	Reader Module	Reader 코드 Read
@G	Data Augmentation	야 너두 데이터 공장 열 수 있어~
@K	fairseq and huggingface	PORORO 모델 포팅기
@U	DPR Module	DPR 특강 (성능 책임 못짐)
@I	Curriculum Learning	기영이의 문제은행
@Y	Data Processing	모델에 들어가기 전과 후가 다른 데이터
@O	Random Masking	마스킹-개념부터 코드까지 핫이슈~

서로 경험하지 못한 부분을 보충하기 위해 대회 종료 후 팀 내부 세미나 진행!

감사합니다! 😊