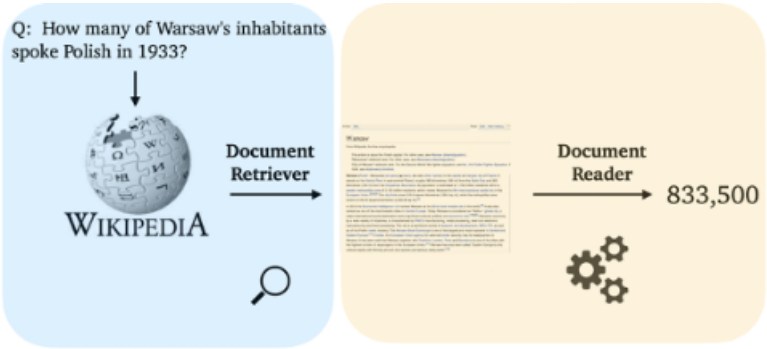
# Machine Reading Comprehension(MRC) 랩업 리포트

| | |
|---|---|
| 생성일 | @2022년 5월 14일 오후 9:20 |
| 참가자 | 차경민 이도훈 오리 Taehun Kim Jinhee Kang |
| 트랙 | NLP |
| 레벨 | Level2 |
| 기간 | @2022/04/25 → 2022/05/12 |

## 🧐 프로젝트 개요

### 프로젝트 개요

> ***ODQA (Open-Domain Question and Answering)***




- 지문이 따로 주어지지 않고 사전에 구축되어 있는 Knowledge resource에서 질문에 대해 대답할 수 있는 문서를 찾은 후, 해당 문서에서 질문에 대한 대답을 찾는 문제
- Query 문장을 받아 2가지 단계의 task를 수행한다.
  1. **Documnet Retriever**
     Query가 주어졌을 때, Query에 대한 대답을 할 수 있는 문서를 문서 사전에서 찾아오는 모델
  2. **Document Reader**
     Retriever을 통해 반환된 문서에서, Query와 가장 관련 깊은 Phase를 찾아 오는 모델

## 프로젝트 팀 구성 및 역할

> 자, 연어 한 접시는 **5명**의 팀원으로 구성되어 있습니다.

#### MRC 프로젝트 팀 구성 및 역할

| member | tag | role |
| --- | --- | --- |
| 김선재_T3252 | BM25 Elastic Search Ensemble Model Optimization | BERT-base 모델 실험, RoBERTa-large 모델 실험 및 최적화, Sparse-retriever TFIDF, BM25 실험, Elastic Search 실험, ensemble 실험 |
| 차경민_T3215 | Augmentation Dense Retriever EDA Model preprocessing | RoBERTa-large 모델 실험 및 최적화, koelectra-base 모델 실험, 데이터 증강, EDA 탐색 및 실험, Dense Retriever 실험, Data preprocessing |
| 이도훈_T3140 | BM25 Dense Retriever EDA Ensemble Model Optimization | RoBERTa-large 모델 실험 및 최적화, xlm-RoBERTa-large 모델 실험, EDA 탐색 및 실험, BM25 실험, Data preprocessing, ensemble 실험 |
| 김태훈_T3065 | EDA Question Generation management | EDA, github repository 관리, pre-trained model 탐색, huggingface 모델 실험 환경 셋팅, Question Generation 구현 및 실험 |
| 강진희_T3007 | EDA Model Optimization Sparse Retriever | RoBERTa-large 모델 실험, koelectra-base 모델 실험 및 최적화, EDA 탐색 및 실험, ensemble 실험, Sparse-retriever TFIDF, BM25 실험 |

## 프로젝트 수행 절차 및 방법

### 🕑 프로젝트 타임라인

1주차 **EDA**, **Augmentation**, **Search**

- 자율적인 실험 아이디어 공유 및 실험 계획과 수행, 베이스라인 학습 및 데이터 증강, 실험 환경 세팅

2주차 **Model**, **Optimization**,

- 모델 아키텍처 선택 및 결과 공유, 모델 최적화, Retriever, Preprocessing, Elastic search 실험

3주차 **Optimization**, **Ensemble**

- Retriever, Preprocessing, Elastic search 실험, 모델 최적화 및 앙상블 수행

---

🗣️ 협업 방식

- 모델 학습 코드, 앙상블 생성 코드 등 정리한 코드는 github 레파지토리에 공유한다.
- 각자 실험이 있으면 따로 branch를 만든 후, 실험이 끝나면 master branch에 pull request를 통해 merge 시킨다.
- 모델 학습 시 wanDB 공통 프로젝트에 기록하며, run name은 모델명과 기타 실험 옵션이 드러나도록 설정한다.
- wanDB 기록 시, tag 사용 및 노트 사용을 적극적으로 활용하여 실험 공유가 용이하도록 기록한다.
- 오전 10시 데일리 스크럼에서 이전까지의 실험 결과와 이후 이어갈 실험 계획 혹은 새로운 실험 아이디어를 공유한다.
- 오후 4시 피어세션에서 오전-오후에 수행한 실험 결과를 공유하고, 팀원의 아이디어 및 실험에 피드백을 제공한다.

- 추가적 논의는 오후 9시 이후 게더타운 혹은 메신저를 통해 공유한다.

## 프로젝트 수행 결과

### 📌 Data Augmentation

- 주어진 Question, Context의 데이터 사이즈가 다른 pre-train에 사용되는 다른 자연어 데이터 셋에 비해 규모가 작다고 판단
- KorQuAD 데이터 셋을 기존에 데이터 셋에 추가하였지만 효과가 없었음. 그래서 back-translation을 통한 데이터 증강을 시도하였지만 오히려 성능이 떨어짐.

### 📌 Preprocessing

- 제공 받은 Wikipedia 데이터 셋에는 여러가지 특수 문자들이 섞여 들어가 있다. 이 중 정답으로 쓰이지 않는 특수 문자는 대부분 Retrieval의 성능에 noise를 줄 것이라고 판단하였다.
- '\n', '#', 한자 등 정답에 해당하지 않는 문자들을 제거하였다.
- 기존의 모델보다 약 1~2%의 성능 향상

### 📌 BM25

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) * \frac{f(q_i,D) * (k_1+1)}{f(q_i,D) + k_1 * (1 - b + b * \frac{|D|}{avgdl})}$$

$f(q_i,D)$: 문서 $D$에서 $q_i$의 term frequency
$|D|$: 문서 $D$의 길이
$k_1$, $b$: 파라메터
$avgdl$: 문서 집합의 평균 문서 길이

BM25 score 점수 계산 방식 : https://littlefoxdiary.tistory.com/12

- 주어진 쿼리에 대해 문서의 연관성을 평가하는 랭킹 함수
- TF-IDF의 개념을 바탕으로, 문서의 길이까지 고려하여 scoring 하며, TF 값에 한계를 지정해두어 일정한 범위를 유지하도록 한다.
- 파이썬의 rank_bm25를 사용하여 구현하였으며 기존 모델보다 EM기준 15점 정도 성능 향상이 있었음
- top-k 별로 문서 accuracy를 정리하지 못한 것 은 아쉬웠다.

### 📌 Elastic Search

- 선재님꺼 보고 적기. 컨닝페이퍼 https://naem1023.notion.site/ODQA-4be47dae144f479fb70431181cdd1cbc
- elastic/elasticsearch-py: Official Elasticsearch client library for Python (github.com)

  thejungwon/search-engine-tutorial (github.com),

  https://stages.ai/competitions/192/discussion/talk/post/1342

  을 보며 구현을 시도했다.
- 검색 엔진 세팅을 하는 데까지 성공했으나 Elastic Search에 대한 이해 부족과 시간 부족으로 인해 Retrieval 코드에 제대로 적용을 하지 못했다.

📌 **Dense Retrieval**

- Passage와 Question을 각각의 pretrained model을 만들어서 hidden representation 벡터들의 내적을 통해 답을 얻을 수 있는 passage와 question의 내적 값을 높이는 방향으로 학습 시키는 방법
- 단어의 유사성 혹은 맥락 해석이 가능하고, 차원이 sparse retrieval 보다는 상대적으로 작아서 다양한 방법론에 적용 가능하고 효율적이라 생각
- “Dense Passage Retrieval for Open-Domain Question Answering 논문에서 제시한 in-batch negative sampling 기법을 통해 Dense Retrieval을 구현하려 하였으나 Dataset size의 한계와 GPU resource의 제한으로 논문에서 제시한 batch size로는 재현하지 못함.

## 자체 평가 의견

👍 **잘한 점**

- (협업) 기존에 Github으로 코드 공유, 데이터 공유만 하던 것을 넘어서 버전 관리를 활용하였다. 각 실험마다 새로운 브랜치를 만들고 실험이 성공적으로 끝나면 master 브랜치에 pull request를 하는 방식으로 진행하였다.
- (협업) Github 에서 Commit 메시지 네이밍 규칙을 정하여 한 눈에 어떤 작업을 하였는지 알 수 있었다.
- ODQA에 알맞은 retriever(elastic search, DPR bm25)를 찾아 적용 시켜본 점

💡 **시도했으나 잘 되지 않은 것**

- 파이썬 코드의 모듈화에 대한 이해 부족과 전체 베이스 코드 라인 이해의 부족으로 Dense Retriever 구현을 시도했으나 data batch 사이즈의 한계로 인하여 적용하지는 못함.
- 데이터 전처리, 후처리를 진행하였지만 좋은 결과로 이어지지는 않았다.

✏️ **아쉬운 점**

- 이전 대회들 보다 베이스 코드가 어려워서 코드를 이해하기 힘들었다. 대회 중간에 베이스 코드를 조금이라도 뜯어보는 시간을 가지기는 하였지만 처음부터 시작했으면 어땠을까 하는 생각이 들었다.
- 대회 초반에 실험 환경을 세팅하지 못한 점
- 깊고 꼼꼼한 실험을 진행하지 못한 점

---

**개인 회고**

**이도훈_T3140**

- [학습 목표]
  - 이번 진행한 프로젝트는 기계독해 프로젝트로, Open-domain Q&A였다. Q&A라는 새로운 NLP task에 도전하는 만큼 또 다른 방법론들을 배우고 실습해볼 좋은 기회라고 생각했다. 이번 프로젝트는 Retriever와 Reader 두 가지 step으로 구성된다. 먼저 Retriever 분야만으로도 상당히 많은 내용들을 배울 수 있었다.
  - Sparse embedding과 Dense embedding의 방법론 및 장단점 실습
  - Extraction based model과 Generation based model의 비교
- [시도해 본 것]

- Sparse embedding: 구현하는 방법에는 TF-IDF와 TF-IDF의 개선인 BM25가 있다. 이번 프로젝트를 하면서 베이스라인에 적용되어 있는 TF-IDF를 BM25로 개선시켜 보았는데, 이것 만으로 EM이 36에서 47로 상당히 크게 증가했다. BM25에서는 문서의 길이가 일정 이상 길어지는 경우 TF가 수렴하도록 한다. 이를 통해서 긴 문서 때문에 단어 빈도수가 계속해서 증가하는 것을 막아 TF의 영향력이 지나치게 커지는 것을 막는다고 한다.
- Dense embedding: 일반적으로 Sparse embedding과 Dense embedding을 같이 사용하거나, Dense embedding을 사용하는 것이 더 성능이 좋다고 한다. 그러나 이번 프로젝트에서 Dense embedding을 구축하여 비교했는데 EM이 47.08에서 48.10으로 약간 증가하기는 했지만 크게 증가하지는 않았다. 아무래도 대회 막바지 즈음에 시도를 해서 최적의 학습을 하지 못 했기 때문이라고 생각한다.
- 마지막으로 Reader의 경우에는 KoElectra와 Bert, RoBERTa 등 여러 extraction base 모델을 시도해보았다. 이중에서는 역시 RoBERTa가 제일 성능이 좋았다.

- [아쉬운 점]
  - 전체적으로 새로운 것들을 많이 배워서 꽤 낯설고 어려운 대회였다. 강의와 과제를 통해 어느 정도 배우긴 했어도 모르는 부분이 많아 시간이 많이 걸렸다. 특히 Elastic Search 같은 경우에는 아예 고려조차 못 해봐서 많이 아쉬웠다.
  - 또 한가지 아쉬운 점은 generation base 모델을 시도해보지 못 했다는 점이었다. 기존 passage에서 정답을 잘 찾아내는 것도 꽤 흥미로운 주제였지만, passage를 학습하여 모델이 스스로 정답을 생성해내는 것도 상당히 관심이 있었는데 이번 대회에서는 시간이 부족하여 시도해보지 못 했다. 마침 최종 프로젝트 주제에 생성 모델도 있는 만큼 어떻게 하면 더 자연스러운 생성과 생성 모델의 결과물을 잘 평가할 수 있는지 고민해보려고 한다.

### 차경민_T3215

- 이번 프로젝트에서 나의 목표는 무엇이었는가?
  1. Github 협업 방식에서 버전 관리 시스템 도입
  2. ODQA를 시스템을 제대로 이해하고 실습하기
- 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?
  1. Github 협업에 대한 가이드 라인과 규칙을 만들었다.
     a. 각 실험마다 branch를 만들고 실험을 진행한다.
     b. 실험이 끝나면 master branch로 pull request를 보낸다.
     c. 팀원들이 모두 실험 내용을 확인하면 master 브랜치에 반영
     d. Commit message도 규칙을 만들어 어떤 작업을 했는지 한 눈에 볼 수 있도록 진행하였다.
  2. 피어 세션마다 각 실험 결과를 공유하고 각자 어떤 실험을 할 것인지 역할 분담을 진행하였다.
- 나는 어떤 방식으로 모델을 개선했는가?
  1. Dense Retriever

     Sparse retriever에 비해 단어의 유사성 혹은 맥락 해석이 가능하기에 좋은 성능을 기대하였다. 하지만 논문에서 제시한 in-batch negative sampling 기법을 통해 구현을 하려고 하였지만 dataset size의 한계와 GPU resource의 제한으로 논문에서 제시한 batch size로는 재현하지 못함.
  2. Preprocessing

데이터 셋 안에서의 줄바꿈 문자, 한자와 같이 pretraining 할 때 불필요할 것 같다고 생각되는 문자들을 전처리 하는 작업을 수행하였다. 성능이 미세하게 오름.

3. 데이터 증강

   korQuad 데이터 셋을 back-translation을 통해 데이터 증강하였지만 성능이 오히려 떨어짐.

- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

  이번 대회는 대부분 모든 실험이 실패였다. 특히 dense retrieval을 구현하는 과정이 너무나 힘겨웠었다. 대회를 마치고 이전 기수들의 코드들을 보니 각 기능들을 패키지로 만들어서 어느 코드에서나 패키지를 가져다 쓸 수 있게 작성된 코드를 보고 살짝 벽을 느꼈다. 주말 동안에 허깅 페이스에서 Trainer 클래스 부분과 파이썬 모듈화 부분을 공부해야겠다는 생각을 하였다.

- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

  이번 대회는 이전 대회에 비해서 협업이 잘 이루어졌던 것 같다. 초반에는 조금 복잡한 점이 많았지만 각 기능을 버전 별로 관리하니 코드 공유도 그렇고 협업하는 것에 있어서 많은 도움이 되었다.

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

  각 기능들을 패키지화 해서 구현했으면 어땠을까 싶다. 그냥 베이스 코드를 그대로 사용하였는데 대회 초반에 하이퍼 파라미터 변수나 기능들을 쉘이나 패키지로 세팅했으면 후반에 더 수월하지 않았을까 하는 아쉬움이 들었다.

- 한계/교훈을 바탕으로 다음 p-stage에서 스스로 새롭게 시도해볼 것은 무엇일까?

  노션에 각 실험들을 정리하고 왜 시행했는지. 잘 된 이유는 무엇이고 안된 이유는 무엇인지 작성해보면 더 좋을 것 같다.

**김선재_T3252**

1. 학습 목표
   - 질문에 관계된 문서를 찾아주는 Retrieval 단계, 문서에서 적절한 답안을 만들고 찾아주는 Reader 단계를 이해하는 것
   - 아이디어를 공유하고 이에 대한 피드백을 받는 것
2. 시도해 본 것
   - Retrieval 단계: Elastic Search

     전에 공부하며 찾은 elastic search 공식 문서 자료와

     elastic/elasticsearch-py: Official Elasticsearch client library for Python (github.com)

     slack 채널에서 서중원 조교님이 공유해주신 GitHub 페이지

     thejungwon/search-engine-tutorial (github.com),

     두호 님이 AIstages 게시판에 공유해 주신 내용 aistages 을 보며 구현을 시도했다.

     검색 엔진 세팅을 하는 데까지 성공했으나 떤 키워드로 문장을 검색해야 하는지 제대로 파악하지 못해 Retrieval 코드에 제대로 적용을 하지 못했다.

   - Train 단계: Learning params

     베이스라인 코드에서 기본적으로 제공되는 Arguments들을 살펴보며 현재까지 진행했던 프로젝트를 복기해 보았을 때 효과적으로 작용했던 몇 가지 변수들을 변화시켰다. 대표적으로 weight decay, weight ratio, learning rate, reschedule, best_model_matrix가 모두 초기화 되어 있기에 huggingface에서 모델, 스케쥴러 스펙을 보며 값을 조정해 주었다.

3. 느낀 점
    - RE task와 비슷한 형식의 베이스라인 코드를 가지고 있을 거란 안일한 생각을 가지고 있었다. 시간 분배를 잘 하지 못한 것 같아 아쉬운 감정이 들었다.
    - Elastic Search를 잘 활용하지 못해서 아쉬움이 남았다. 베이스라인 코드에 접목시키기에 시간이 부족해 개인적으로 도커와 함께 더 공부할 예정이다.

**강진희_T3007**

## 학습 목표

- 기계 독해 과제를 이해하고, 적절한 모델을 직접 고민해보고 적용해보기(이 분야에서 자주 쓰는 모델이더라, 베이스라인 모델이라서 썼다→가 아니라 Reader의 역할을 생각해보고 그에 알맞은 모델이 뭔지 고민하는 과정 가지기)
- 베이스라인 코드 직접 해석해보고 커스터마이징 하기

## 수행해본 것

- 베이스라인 코드의 라인by라인 주석 달기 및 학습 결과 찍어보기

## 아쉬운 점

- 이번 대회는 스스로에게 무척 실망스러웠다.대회 전반의 참여도, 기여도가 없는 수준이다. 이전 대회에서는 적어도 회의록 작성을 통해 팀원들의 실험 흐름을 이해하는 과정을 거쳤는데 이번에는 회의록 작성에도 적극적이지 못했다.모더레이터라는 의무가 할당되어야만 행동하는 수동적인 자세가 아니라, 자발적으로 나서서 나에게 도움이 될 행동을 하고, 그에 따라 팀에 기여하는 행동을 할 줄 알아야 한다.이번 대회의 참여 저조는 나 자신에게 가장 큰 손해이다. 기계 독해 과제 자체가 생소하고, 레퍼런스도 적은 편이고, 어려운 과제라고 생각한다.mrc 이해에 몰두할 수 있는 3주라는 기회가 주어졌지만 이를 적극적으로 활용하지 못한 게 가장 큰 손해라고 생각한다.실험에 적극적으로 참여하지 않은 것, mrc 과제를 공부하고 적절한 모델과 실험을 설계하지 않은 점, 회의록 작성에 적극적으로 참여하지 않았고 이해하려는 의지가 부족했던 것이 아쉽다.

**김태훈_T3065**

# 학습 목표

- 기계독해 프로젝트로, Open-domain QA 태스크를 진행하며 Retriever 와 Reader를 구성하는 스킬에 대해 익숙해지기
- Sparse Embedding 과 Dense Embedding 등 여러 Embedding 방법론의 차이를 경험하고, Extraction-base model, Generative model 적용해보기

# 시도해 본 것

- Retrieval :

Retrieval 과 Reader step 중에서 아무래도 실험적으로 비교할 수 있었던 것은 Retrieval 파트인데, Sparse Embedding 과 Dense Embedding 을 구현해서 비교해볼 수 있었다. 상대적으로 구현이 용이한 Sparse 에 비해

Dense 에서는 Negative Sampling 등을 통해 성능의 개선을 기대했지만, 시간 내 구현의 한계로 만족할만한 성능을 얻지는 못했다. Sparse Embedding 쪽에서 TF-IDF, BM25 등을 비교해보며 진행해본 정도로 만족해야 했다.

- Reader :

KoELECTRA, RoBERTa 등 잘 알려져있는 BERT 기반의 Extraction-based 여러 모델들을 시험해보는 과정에서, 내부 실험결과가 RoBERTa 가 가장 좋다는 것을 알고 그쪽으로 수렴하게 되었다. Extraction model 과는 별개로 다양한 Generative model 의 실험도 예정에 있었으나 현실적인 한계 때문에 제대로 시도해보지 못했다.

# 느낀점 및 아쉬운 점

- 이번 기간에는 최종 프로젝트도 같이 고민을 하느라, MRC 프로젝트의 경우 성능에 기반해서 의사결정을 했을 뿐, 천천히 숙고하는 과정을 많이 거치지 못해 지나고 나니 큰 후회가 남았다.
- 특히 Generative model 의 경우 최종 프로젝트 주제 안에도 큰 부분으로 포함되어 있어, 많은 인사이트와 실험이 필요한 부분이었는데 일단은 대회 기간 내 제출 때문에 많은 시도를 해보지 못했다.
- 한편 Elastic Search 의 경우 이번에 친숙해지는 것이 목표였으나, 아직 부족한 점이 많은 것 같아 추후에 다시 도전해 볼 생각이다.