KoBigBird 사전학습 모델을 이용한 한국어 MRC

Team 1

양다경, 구선혜, 김유나, 김현수, 김현지, 이용주, 최원서

목차

1. 프로젝트 개요

1. 프로젝트 개요

프로젝트 소개

- 주어진 KLUE-MRC 데이터셋에 대하여 KoBigBird 사전학습 모델을 미세 조정
- 모델이 주어진 질문을 이해하고 뉴스 기사에서 해당 질문에 대한 답을 찾을 수 있음

프로젝트 목표

- 테스트 데이터셋에 대해 Levenshtein distance 1.71678 이하

사용 스택

- Google Colab Pro+
- datasets-2.8.0
- levenshtein-0.20.9
- matplotlib-3.2.2
- numpy-1.21.6
- pandas-1.3.5
- transformers-4.25.1
- torch-1.13.0
- torchinfo-1.7.1
- wandb-0.13.7

2. 프로젝트 팀 구성 및 역할

2. 프로젝트 팀 구성 및 역할

양다경

- 베이스라인 코드 분석
- 코드 수정
- 외부 데이터 추가
- 훈련 및 검증 데이터 뉴스 카테고리 EDA
- 모델 검증
- 에러 케이스 EDA
- 후처리
- 프로젝트 결과물 작성

구선혜

- 베이스라인 코드 분석
- 적용할 수 있는 SOTA 모델 조사
- 아키텍처 측면에서 OOM 문제 개선 및 훈련 속도 개선 방안 검토
- 에러 케이스 분석

김유나

- 베이스라인 코드 분석
- 외부 데이터 추가
- 훈련 및 검증 데이터 뉴스 카테고리 및 답변 길이 EDA
- 훈련 속도 개선 방안 조사 (fp16)
- 모델 훈련

2. 프로젝트 팀 구성 및 역할

김현수

- 베이스라인 코드 분석
- 컴퓨팅 단위 소진 문제 조사
- 발표

최원서

- 베이스라인 코드 분석
- 자료 조사 (BigBird)

김현지

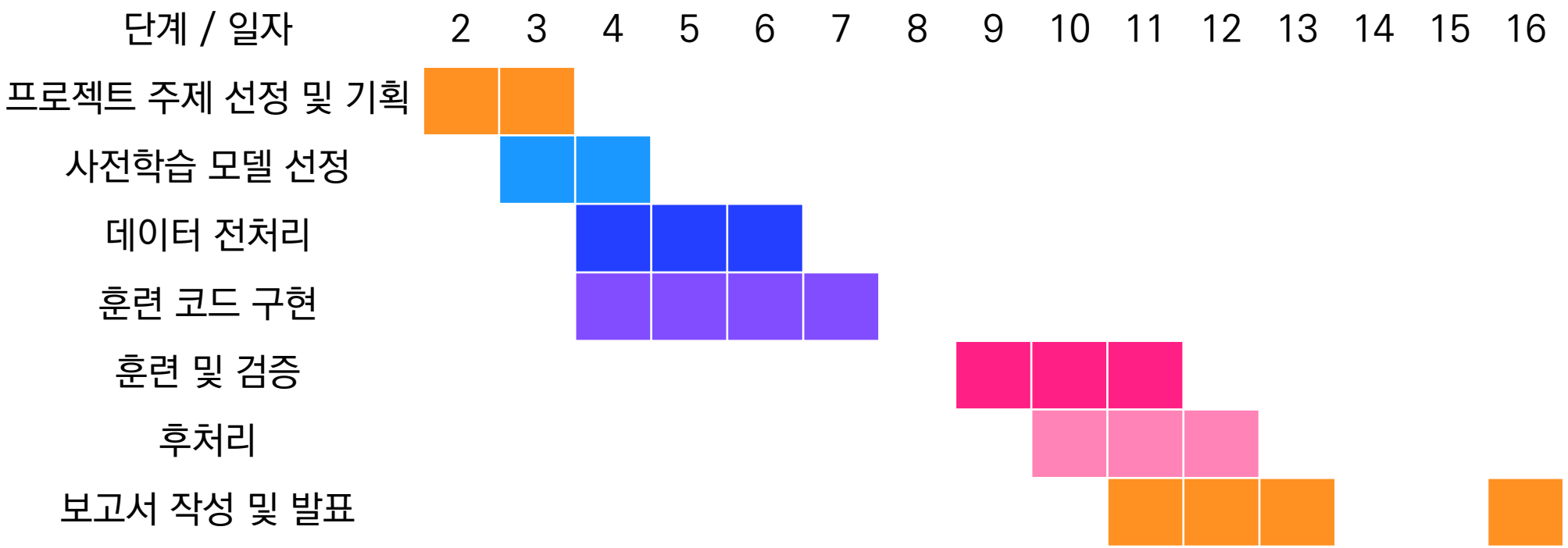
- 베이스라인 코드 분석
- 자료 조사 (KoBigBird, CosineAnnealingLR)

이용주

- 베이스라인 코드 분석
- 훈련 데이터 tokenization EDA
- OOM 문제 개선 방안 검토 (accumulation)
- 후처리

3. 프로젝트 진행 프로세스

3. 프로젝트 진행 프로세스

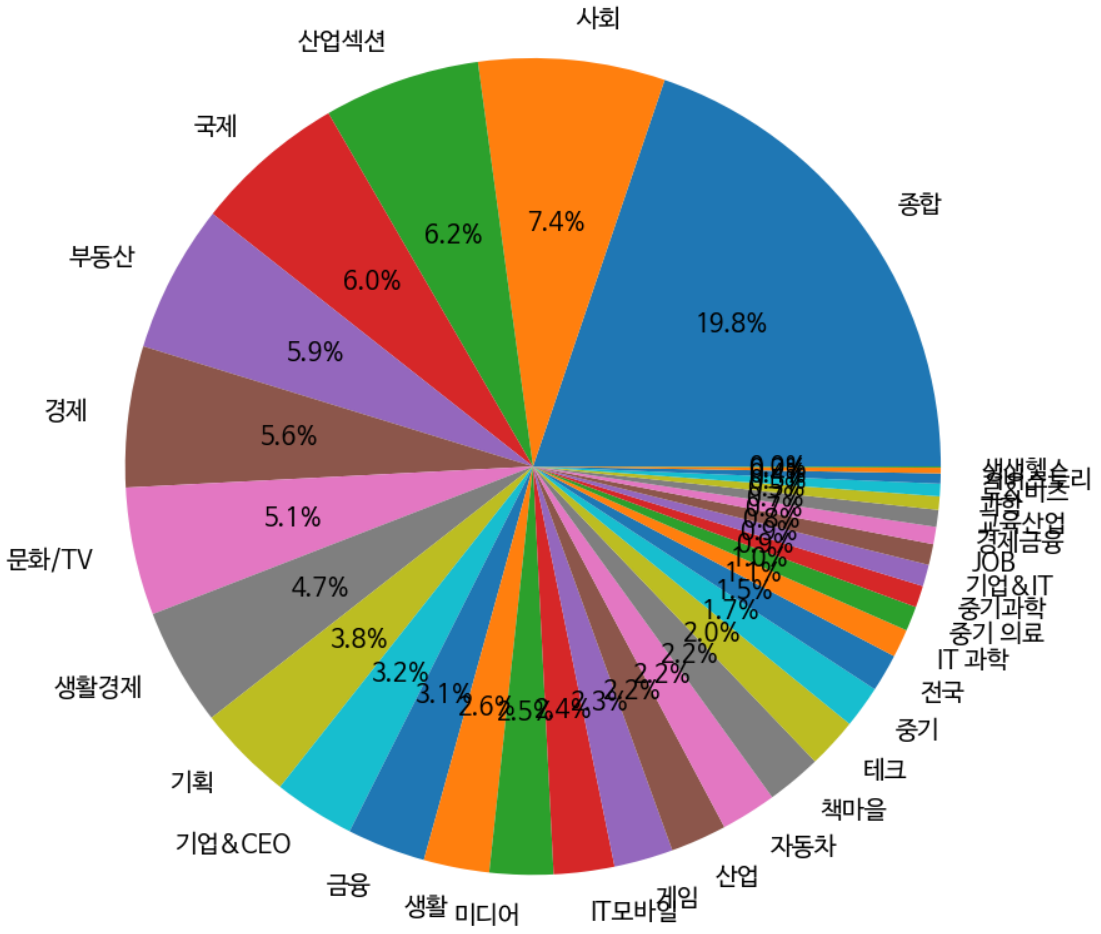


4. 프로젝트 결과

1. 전처리
2. 모델 훈련 및 검증
3. 후처리

4-1. 전처리
- 훈련 및 검증 데이터셋

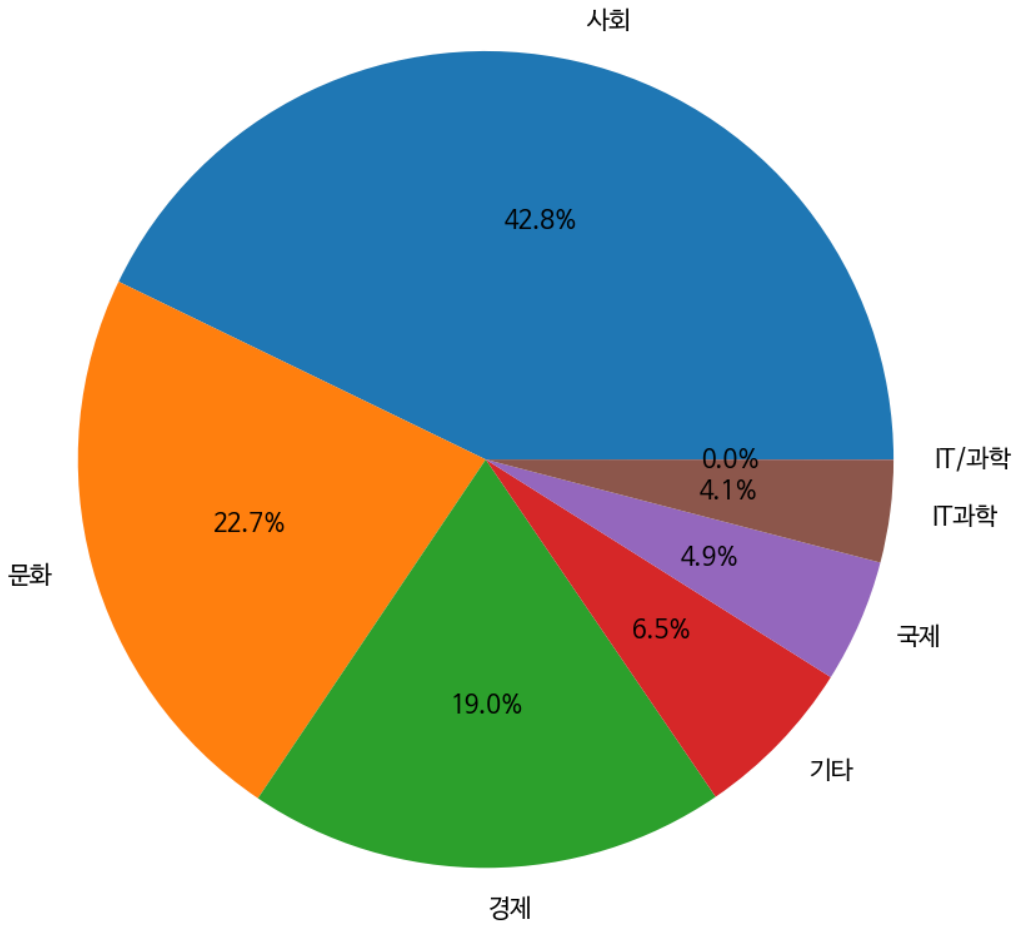
- KLUE MRC 데이터셋
  - 12,037 개 샘플
  - 8:2의 비율로 데이터셋 분리 (무작위 추출)
    - 훈련 데이터셋: 9,630 개
    - 검증 데이터셋: 2,407 개



〈KLUE MRC 데이터셋의 뉴스 카테고리 분포〉

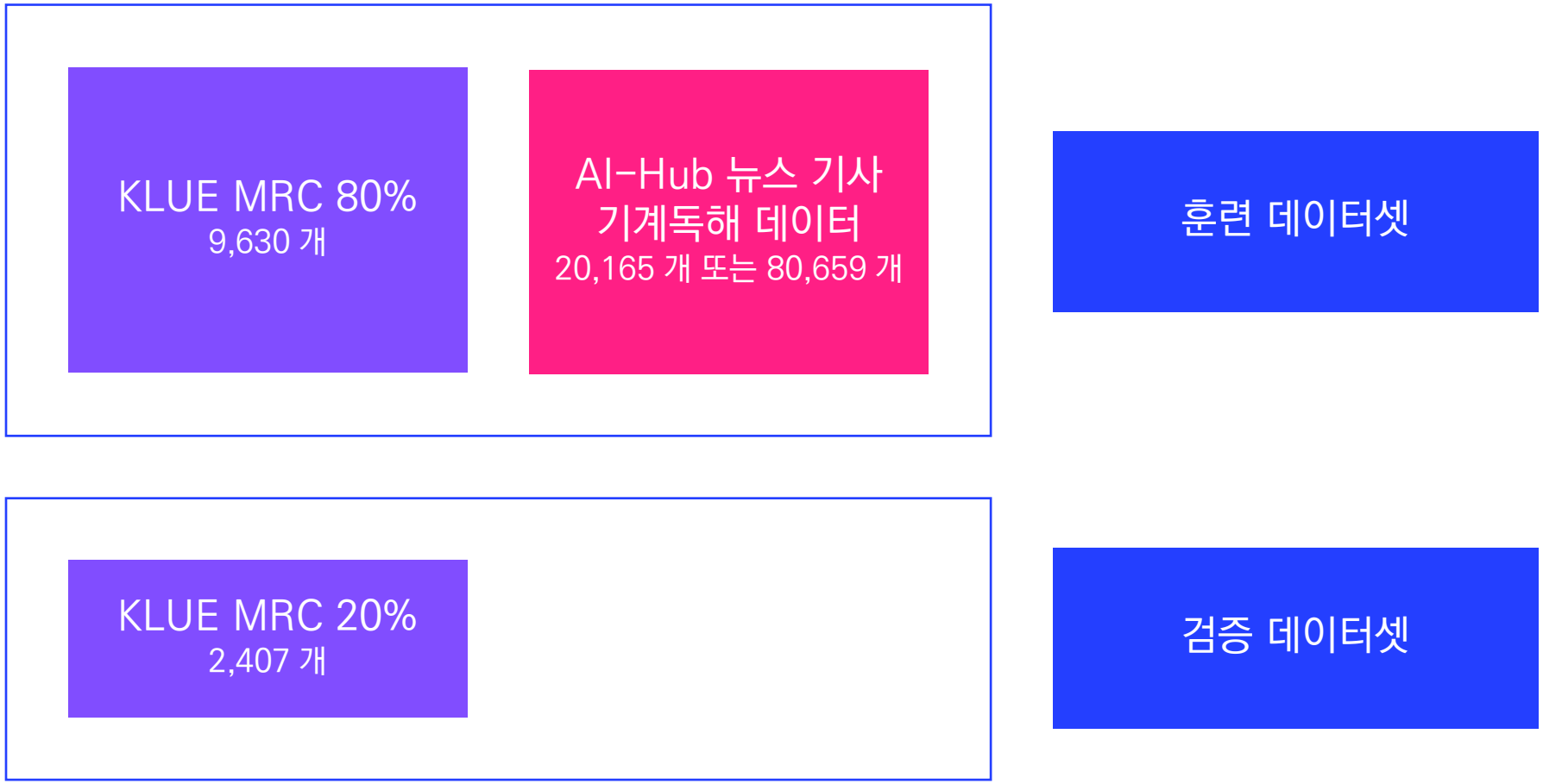
4-1. 전처리
- 훈련 및 검증 데이터셋

- AI-Hub 뉴스 기사 기계독해 데이터
  - Training 라벨링데이터 中 추출형(서술형)
  - 카테고리가 정치, 지역, 스포츠인 경우 제외
  - 15% 무작위 추출
    - 20,165 개 샘플
  - 60% 무작위 추출
    - 80,659 개 샘플
  - 훈련 데이터셋에 추가



〈AI-Hub 뉴스 기사 기계독해 데이터의 뉴스 카테고리 분포〉
(정치, 지역, 스포츠 제외)

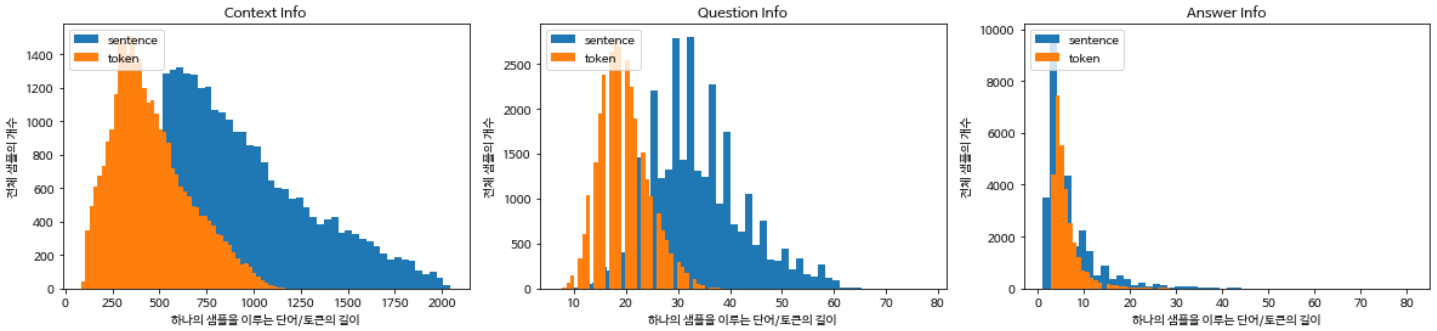
4-1. 전처리 - 훈련 및 검증 데이터셋



4-1. 전처리 - Tokenization

〈훈련 데이터셋의 Context, Question, Answer의
문장 길이와 토큰 수 요약 통계〉

	Context		Question		Answer	
	문장 길이	토큰 수	문장 길이	토큰 수	문장 길이	토큰 수
max	2046	1166	78	45	81	49
min	179	84	8	7	1	3
mean	874.33	454.36	33.49	19.57	6.93	5.95
std	394.98	204.93	9.06	4.62	5.96	3.19
median	802	416	33	19	5	5
Q1	578	301	27	16	3	4
Q3	1118	578	39	22	8	7



- BertTokenizer 사용
- max_length: 1024
- doc_stride: 200
- padding: “max_length”

4. 프로젝트 결과

1. 전처리
2. 모델 훈련 및 검증
3. 후처리

4-2. 모델 훈련 - 사전학습 모델 선정

- KoBigBird

  - Released for long-rage under-standing of Korean language
  - Covers with more than 8 times longer than the usual (512 tokens) BERT models
  - KoBigBird scored the highest score (87.08 EM score / 94.71 F1-score)

Kichang Yang. Transformer-based Korean Pretrained Language Models: A Survey on Three Years of Progress.

TABLE 2
A Result of Multiple Sentence & Agent Tasks.

Models	KorNLI[1]	KorSTS[2]	Question Pair[3]	KorQuaD (Dev)[4]	Size (MB)
KoELECTRA (Small)	78.6	80.79	94.85	82.11 / 91.13	54
KoELECTRA (Base)	**82.24**	**85.53**	95.25	84.83 / 93.45	431
DistilKoBERT	72	72.59	92.48	54.40 / 77.97	108
KoBERT	79.62	81.59	94.85	51.75 / 79.15	351
SoongsilBERT (Small)	76	74.2	92	-	213
SoongsilBERT (Base)	78.3	76	94	-	370
KcBERT (Base)	74.85	75.57	93.93	60.25 / 84.39	417
KcBERT (Large)	76.99	77.49	94.06	62.16 / 86.64	1200
KoBigBird (Base)	-	-	-	**87.08 / 94.71**	436
KoBART	-	81.66	94.34	-	473
KoGPT2	-	78.4	-	-	490
HanBERT	80.32	82.73	94.72	78.74 / 92.02	614
XLM-Roberta-Base	80.23	78.45	93.8	64.70 / 88.94	1030
KcELECTRA-base	81.65	82.65	**95.78**	70.60 / 90.11	475

[1, 3] measured by accuracy.
[2] measured by spearman correlation.
[4] measured by (1) EM score and (2) F1 score.

4-2. 모델 훈련 - Training Arguments

- train_batch_size: 512
- gradient_accumulation_steps: 32 → OutOfMemory 문제 개선
- fp16: True → 훈련 속도 개선
- train_epochs: 30
    - early_stopping_patience: 5
- tokenizer: BertTokenizer

4-2. 모델 훈련 - Training Arguments

- optimizers
  - optimizer: AdamW
    - lr=1e-1 또는 1e-5
    - weight_decay=0.01
  - scheduler: cosine_schedule_with_warmup
    - num_warmup_steps=2000
    - num_training_steps=60000

※ CosineAnnealingLR

- 매우 큰 학습률로 시작한 다음 다시 학습률을 높이기 전에 0에 가까운 값으로 공격적으로 줄이는 스케줄링 기법

  CosineAnnealingLR

- AdamW가 Adam보다 모든 LearningRateScheduler에 대해 좋은 성능을 얻었는데, 특히 CosineAnnealingLR를 사용했을 때의 성능이 더 좋았다.

  Ilya Loshchilov & Frank Hutter. DECOUPLED WEIGHT DECAY REGULARIZATION.

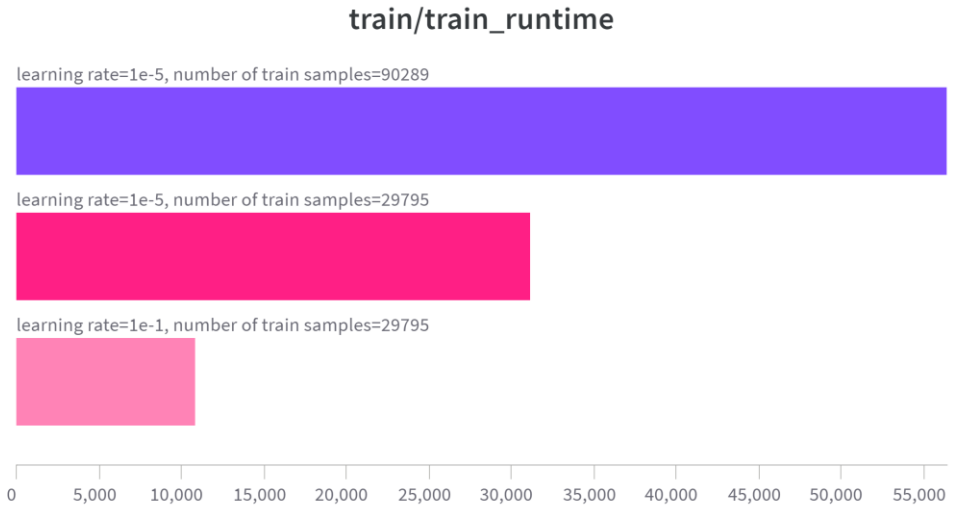
4-2. 모델 훈련 – 결과

	Number of train samples	Learning rate		Train runtime (시간)	Exact match	F1	Levenshtein distance*
Model 1	29,795	1e-1		3.0	0.08309	0.30355	6.27545
Model 2	29,795	1e-5	→	8.7	65.76651	71.40272	1.62775
Model 3	90,289	1e-5		15.7	67.55297	72.93713	1.58828

〈Number of Train Samples와 Learning Rate에 따른
Levenshtein Distance, Exact Match, 그리고 F1 Score〉

* 정답이 여러 개 존재할 경우 제일 길이가 짧은 정답과의 편집 거리

4-2. 모델 훈련 – 결과 (WandB)



➡ 학습률 1e-1에 비해 1e-5는 안정적으로 loss 값이 줄어듦

4. 프로젝트 결과

1. 전처리
2. 모델 훈련 및 검증
3. 후처리

4-3. 후처리 (1)

- n_best_size
  - 예측할 수 있는 답변의 개수
  - 후보: 10, 20, 30
- max_answer_length
  - 생성할 수 있는 최대 답변의 길이
  - 후보: 3, 5, 7, 10, 20

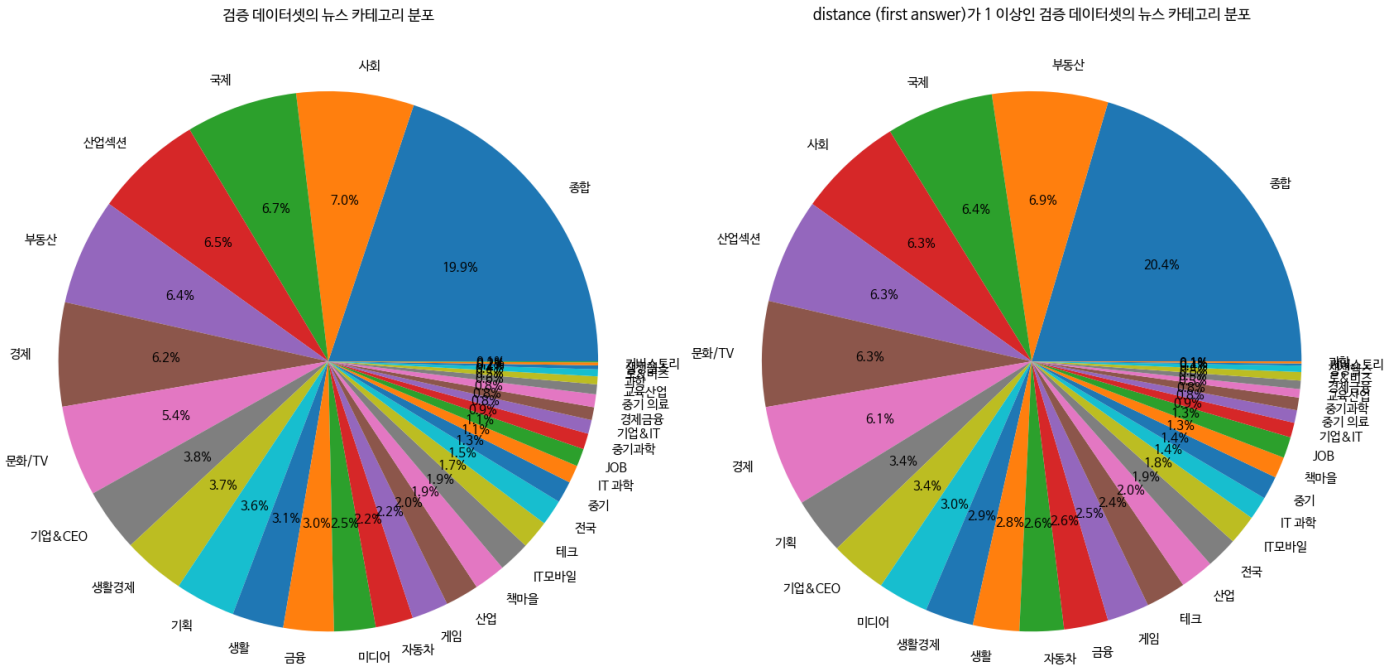
⬇

- n_best_size = 10, max_answer_length = 10인 경우가 가장 Levenshtein distance가 낮음

〈n_best_size와 max_answer_length에 따른 Levenshtein Distance, Exact Match, 그리고 F1 Score〉

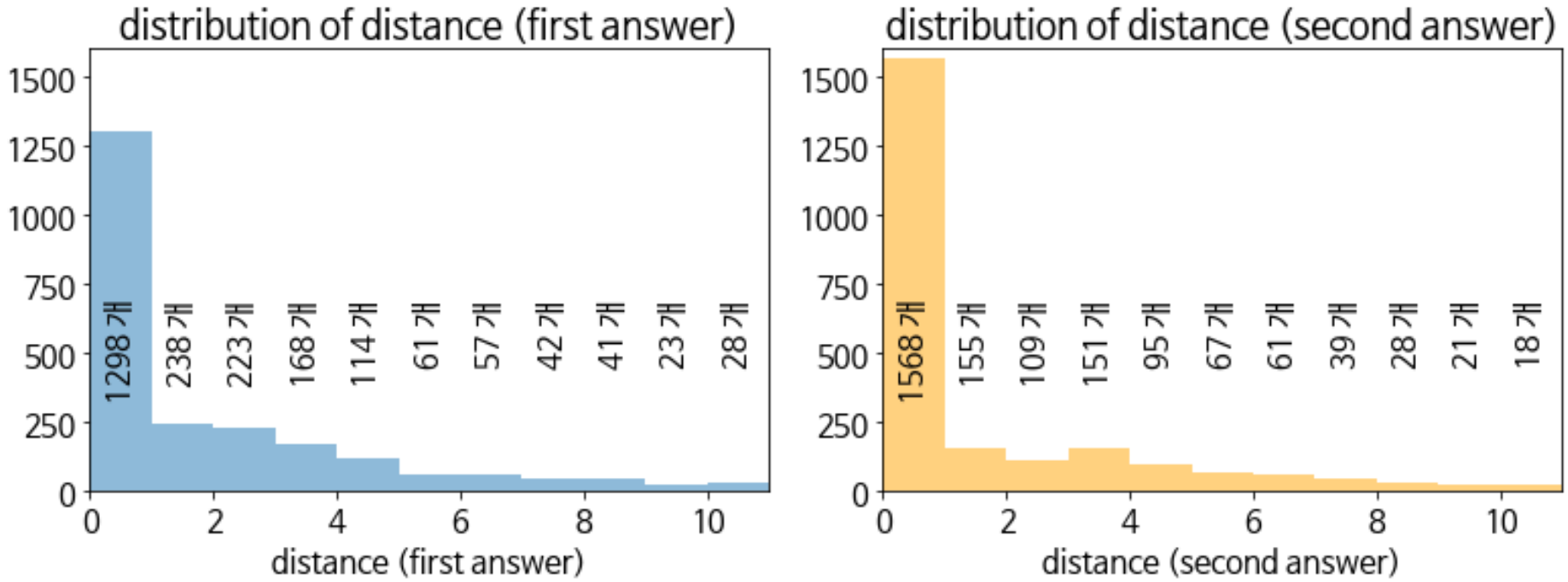
n_best_size	max_answer_length	Levenshtein distance (첫 번째)	Levenshtein distance (두 번째)	Exact match	F1
10	10	2.22227	1.79560	67.55297	72.93713
20	10	2.22227	1.79560	67.55297	72.93713
30	10	2.22227	1.79560	67.55297	72.93713
10	7	2.28625	1.81263	67.05442	72.61127
20	7	2.28625	1.81263	67.05442	72.61127
30	7	2.28625	1.81263	67.05442	72.61127
10	20	2.38637	1.96261	67.09597	72.73440
20	20	2.38637	1.96261	67.09597	72.73440
30	20	2.38637	1.96261	67.09597	72.73440
20	5	2.43249	1.91649	64.85251	70.84363
30	5	2.43249	1.91649	64.85251	70.84363
10	5	2.43581	1.92023	64.85251	70.81198
30	3	3.04321	2.44038	55.62941	63.12170
20	3	3.04445	2.44163	55.62941	63.10508
10	3	3.05152	2.44994	55.67096	63.12190

4-3. 후처리 - 에러 케이스 분석



뉴스 카테고리 간 성능 차이는 없음

4-3. 후처리 - 에러 케이스 분석



첫 번째 정답과 두 번째 정답의 Levenshtein distance 차이가 있음

e.g., 예측한 답변: '11' / 첫 번째 정답: '11월' / 두 번째 정답: '11'

두 번째 정답과는 distance가 0이지만, 첫 번째 정답과는 1

4–3. 후처리 (2) – 형태소 분석

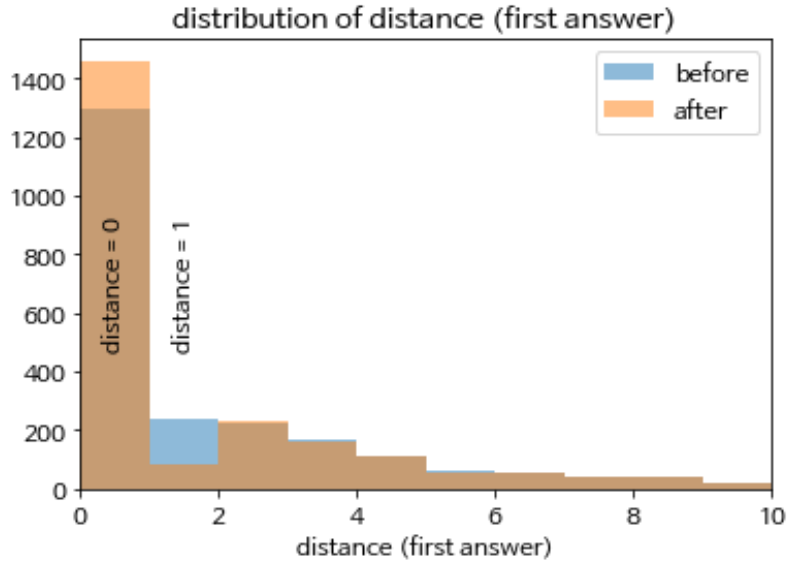
- Khaiii 형태소 분석기 이용
- 예측한 답변 뒤에
  1. 의존 명사 & 기타 기호(논리, 수학 기호, 화폐 기호 등) 추가
  2. 조사 제거

의존 명사 & 기타 기호 추가		조사 제거		의존 명사 & 기타 기호 추가 (잘못된 경우)		조사 제거 (잘못된 경우)	
전	후	전	후	전	후	전	후
15	15년	홈택스로	홈텍스	일본	일본뿐	중저가	중저
1.8	1.8%	20여 명의	20여 명	15년	15년간	손정의	손정

4-3. 후처리 (2) - 형태소 분석

- 첫 번째 정답에 대한 Levenshtein distance의 경우
  ➡ 0.08558 감소
- 두 번째 정답에 대한 Levenshtein distance의 경우
  ➡ 0.04487 증가

➡ Kaggle leaderboard score를 올리기 위해
테스트 데이터셋에도 형태소 분석을 이용한 후처리 적용



Levenshtein distance (첫 번째 정답)			Levenshtein distance (두 번째 정답)		
전	후	차이	전	후	차이
2.22227	2.13668	-0.08558	1.79560	1.84047	0.04487

5. 자체 평가 및 보완

5. 자체 평가 및 보완

자체 평가

- 주어진 KLUE-MRC 데이터셋에 대하여 KoBigBird 사전학습 모델을 미세 조정하여 검증 데이터셋에 대해 Levenshtein distance 2.13668 (첫 번째 정답) & 1.84047 (두 번째 정답)의 성능을 보임
- Kaggle competition 최종 public score 2.14597로, 목표했던 점수에는 미치지 못하지만, 베이스라인 코드의 public score 194.29631에서 크게 향상됨

보완

- Computing unit 부족 문제로 인해 다양한 모델을 훈련할 수 없었음
- 훈련 속도를 크게 향상하는 요인 탐색에 어려움을 겪음
- 제한된 시간과 자원으로 hyperparameter tuning을 정교하게 수행하지 못함

감사합니다.

GitHub: https://github.com/yangdk02/Goorm-AI-NLP-6

References:

- https://github.com/huggingface/notebooks/blob/main/examples/question_answering.ipynb
- https://huggingface.co/monologg/kobigbird-bert-base