NLP-04

Machine Reading Comprehension

PRESENTED BY:

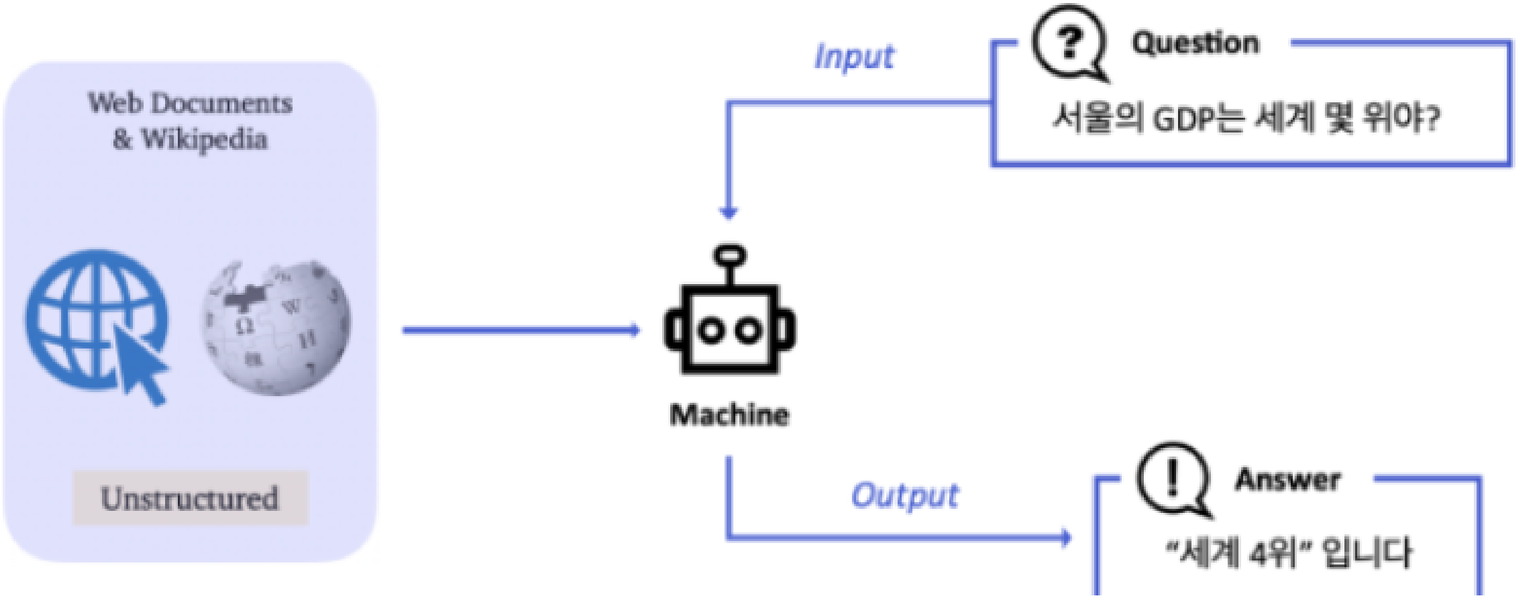
SON YOORIM

Contents

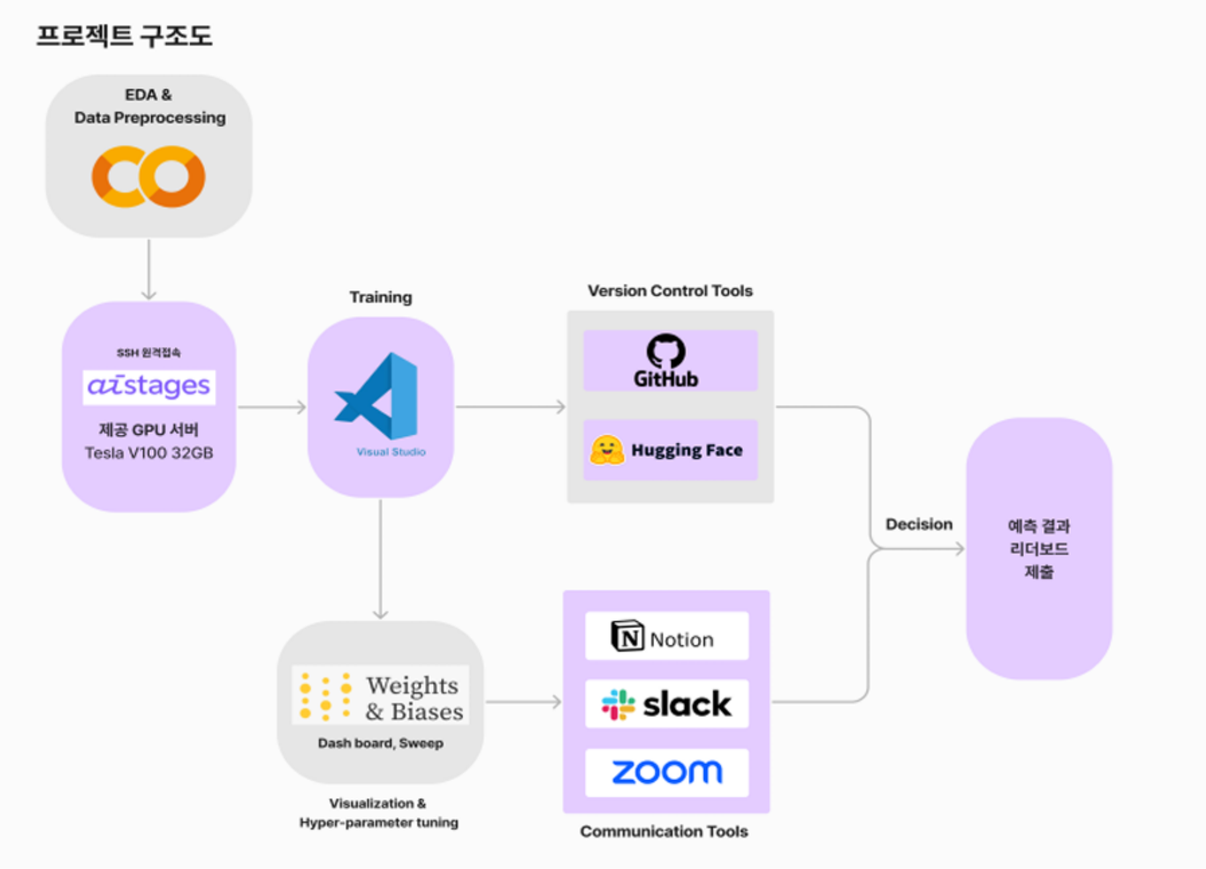
대회 목표

Linking MRC and Retrieval: Open-domain Question Answering (ODQA)

ODQA: 지문이 따로 주어지지 않음. 방대한 World Knowledge에 기반해서 질의응답



개발 및 협업 환경



- Github
  - 단위 Task별 issue 발행 & PR
  - Branch : 각자 기능별 branch 생성 후 develop에 merge
- Notion
  - 일간 회의록 / Task Dashboard
  - 데이터셋 및 업무 진행 상황 공유
- WandB
  - Sweep을 이용한 하이퍼 파라미터 탐색
  - 학습 logging

2-Stage Model

Sparse Passage Retrieval

배경

- 단어사전(BoW)을 구축하여 TF-IDF 기반으로 중요도를 계산하여 평가지표 EM의 성능을 높이기 위해 정확한 단어를 비교하기 위한 방법
- TF-IDF 계열 중 SOTA를 달성한 것으로 알려진 BM25 알고리즘을 적용하여 성능 개선을 꾀함

결과

- 대조 결과, 모든 상황에서 유의미한 성능 개선을 보임

Dense Passage Retrieval(DPR)

배경

Dense Embedding은
- 단어의 유사성 혹은 맥락 파악에 유리
- 학습으로 임베딩을 만들어서 추가적인 학습 가능

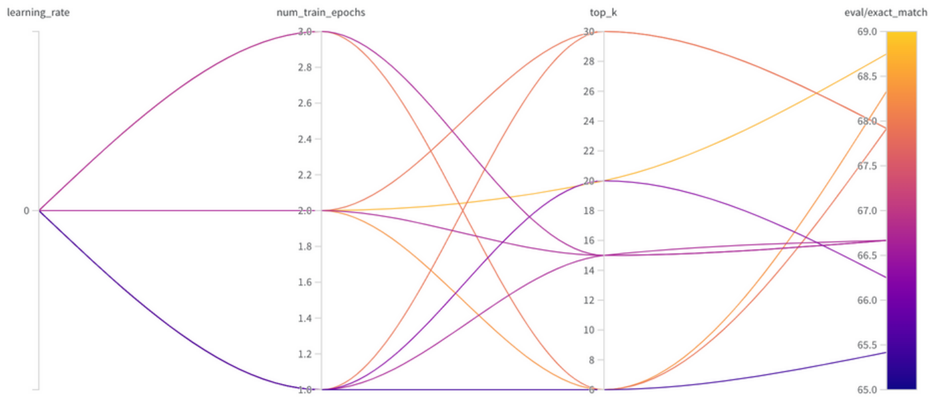
결과

성능 개선 효과를 확인할 수 없었음.

분석

1. 같은 지문에 여러 질의를 가지는 경우 오히려 학습에 방해가 되었다.
2. 시도를 하지 못했지만, 지문단위가 아닌 문장단위로 학습을 했으면 성능이 개선됐었을지도 모른다.
3. 배치 사이즈가 클 수록 학습을 잘하는데 VRAM의 한계로 이는 시도할 수 없었다.
4. 위의 방식을 개선하기 위해, In-batch 방식이 아닌 Pre-batch 방식을 시도했으나 학습을 해도 loss 값이 낮아지지 않는 문제가 발생해서 결과가 좋지 않았다.

PLM + Hyperparameter Tuning



klue/bert-base	raw	dropout = 0.2 (분류기 제외), act = relu	31.6700	45.6400
klue/bert-base	raw	act = relu	33.7500	46.4300
klue/bert-base	raw	epochs = 5	27.0800	38.0900
klue/bert-base	raw	dropout = 0.2, act = gelu_new	35.8300	48.2800
klue/roberta-large	raw	baseline	42.5000	53.8100
klue/bert-base	raw	top-k = 50	30.8300	39.6700

BERT-base, RoBERTa-large 등의 다양한 PLM 을 이용하여 성능 비교.

Wandb 의 Sweep 기능을 사용하여 dropout, random seed, top-k, epochs 등의 하이퍼 파라미터 튜닝을 실험 진행.

KorQuAD 데이터셋 추가

배경

- 기본으로 제공된 학습용 데이터셋은 3952개 밖에 되지 않기에 Reader 모델 성능이 낮다고 판단
- 그 중 대회 데이터셋과 KorQuAD 1.0 데이터셋이 context 길이나 데이터 출처가 유사하다고 파악하여 KorQuAD 1.0 데이터셋을 활용해 데이터 추가하여 학습

방법

- KorQuAD 1.0 데이터셋 중 기존 데이터셋과 길이 분포 맞춰서 filtering하여 60407개의 데이터를 추가하여 학습 Reader 학습 진행

결과

- 기본 베이스라인 Roberta-large 기준 EM 56.6700 → 59.5800으로 성능 상승

KorQuAD Fine-Tuning

배경

- KorQuAD 1.0 데이터셋을 추가하여 학습한 경우, 방법론 조합에 따라 성능 떨어지는 경우도 발생
- pre-trained 모델에 대회 데이터셋을 fine-tuning하는 것보다는, 유사한 korquad 데이터셋으로 fine-tuning 한 후, 대회 데이터셋으로 전이학습을 해주면 성능이 향상될 것이라고 판단하여 2번의 fine-tuning 진행

방법

- korquad 데이터셋의 경우, train 데이터보다 지문 길이가 긴 데이터 존재하므로 해당 데이터 filtering 한 후, roberta-large fine-tuning 진행
- KorQuAD 1.0로 fine-tuning한 모델 불러와서 기존 train 데이터로 한번 더 fine-tuning 진행

결과

- 추론 결과 확인 결과, 제대로 된 단어를 추출하지 않고 조사를 추출. 제대로 학습이 이루어지지 않은 것으로 판단하여 사용하지 않음.

CNN Layer 추가

배경

- 기존 RobertaForQuestionAnswering의 경우 마지막에 Linear layer만을 통과함.
- 그 앞에 CNN layer를 통과해줌으로써 근접 벡터간 연관정보까지 학습되도록 하고자 함.
- 삼성 SDS KorQuAD 1.0 성능평가 솔루션에서 Motivate

방법

Conv1D(K=3) - Conv1D(K=1) - Relu - LayerNormalization으로 구성된 CNN Block을 5개 쌓아줌

결과

EM 62.5000 → 65.4200으로 약 3점 정도의 성능개선

Pre-processing

배경

- 비슷한 키워드가 문서에 여러 번 등장하는 경우, 질문에 맞는 키워드를 잘 추출하지 못하는 것을 발견
- 조사, 어미 등을 제거한 '단어'에 모델이 좀 더 집중할 수 있도록 하기 위해 명사들을 쉼표로 구분해서 question 앞에 추가

방법

- 예시
  대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은? ->
  대통령, 포함, 미국, 행정부, 견제, 국가, 기관, 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?

결과

EM 60.83 → 62.50 성능 향상

Post-processing

배경

- train/validation dataset의 answer 형태와, 모델 추론 결과에서의 answer 형태의 차이점을 줄여 EM 성능 개선을 꾀함

방법

- 일부 특수 기호(<>, 책 제목 기호, 불필요한 말따옴표 등) 제거

결과

- 전반적으로 성능 하락 or 유지
- 모델의 성능이 높아질수록 후처리 후 성능이 떨어진 것을 고려했을 때, 모델 성능이 일정 점수 이상 보장되면 후처리가 불필요하다는 결론을 내림

Predict 파일의 결과물을 기준으로 다수결 hard voting 기법 사용

데이터셋이나 inference 등 모델의 다양성을 고려하여, 리더보드 EM 점수를 기준으로 hard voting 앙상블을 시도하였습니다

46	Finished	@	62.0800	74.5000
45	Finished	@	58.7500	71.0500
53	Finished	⏏	59.5800	69.7100
56	Finished	⏏	65.0000	75.7100
67	Finished	@	67.9200	78.4200

Description

hardvoting_45_46_53_56_67.json

69	Finished	@	70.0000	79.6700

좋았던 점 🙂

- DPR 구현을 위해 관련 논문을 깊게 읽어보며 베이스라인 코드에 대한 이해가 가능했다.
- 깃허브를 참고하며 커스터마이징 하는 과정을 통해 huggingface 모델을 불러오고 수정하는 것에 익숙해졌다.
- 최종 프로젝트에 적용될 수 있는 부분을 찾아가며 대회를 진행하니 공부가 될 수 있어 좋았다.
- 검색 플랫폼에서 사용되는 기술을 직접 구현하며 경험하는 기회가 되어 좋았다.

아쉬웠던 점 😅

- Inference 결과를 사후 분석하고 이를 기반으로 새로운 방법론을 탐색하는 과정이 부족했다.
- Github를 기존보다 조금 러프하게 사용해서 아쉬웠다. 코드가 복잡해지고 실패한 코드를 커밋하려다 보니 체계적인 정리가 미흡했다. 다음에는 기능을 더 상세히 나누어서 작동하는 코드를 세부적으로 정리하고자 한다.
- 해보고 싶은 Task는 많았는데 하나의 Task의 시간이 오래걸려 많이 적용하지 못한 것이 아쉬웠다.

- **CNN Layer 추가** : 삼성 SDS Techtonic 2020 : KorQuAD 1.0 성능개선 Know-how https://www.youtube.com/watch?v=ovD_87gHZO4

- **DPR 구현 관련**:
  - 논문(DPR for ODQA): https://arxiv.org/abs/2004.04906
  - 논문(pre-batch): https://arxiv.org/abs/2012.12624
  - 프레임워크(haystack) : https://haystack.deepset.ai/tutorials
  - 깃허브(facebookresearch) : https://github.com/facebookresearch/DPR
  - 깃허브(이전 기수) : https://github.com/boostcampaitech5, ...
  - 미션, 실습(부스트캠프 AI Tech MRC 강의)

왕감사합니다