

# 모델링 및 평가

## (1) 기업 분석 보고서 (Domain Name : Company\_Report)

1. 기업의 개요    당사의 주요 사업은 소매유통업으로 편의점 GS25, 슈퍼마켓 GS THE FRESH, 홈쇼핑 GS SHOP 등을 운영하고 있으며, 부동산 관련 개발 기획, 인테리어, 자산관리 및 매각 등 부동산 개발과 상업시설 운영을 전문으로하는 부동산 개발업의 확충을 위하여 운영하고 있습니다. 당사의 주요 종속회사인 파르나스호텔은 우수한 입지 및 높은 대외 인지도를 바탕으로 그랜드 인터컨티넨탈, 코엑스 인터컨티넨탈 및 프렌치 오피스 파르나스 타워 등을 운영하고 있으며, 2017년 11월 불우의 전문성과 경쟁력 강화를 위해 “불우 전문회사 설립”을 이사회에서 결정하고 2018년 1월 불우전문회사 “㈜파르나스호텔”을 설립하였습니다. 향후 경쟁력 강화를 위해 2020년에는 축산물의 제조, 가공 및 유통을 전문으로 하는 ㈜파르나스마트를 설립하였습니다. 2021년에는 농산물의 가공, 유통 및 판매를 주요 목적으로 하는 ㈜파르나스마트의 지분 90%를 취득하였으며, 2023년 12월 중 추가 지분을 취득하여, 2023년말 ㈜파르나스마트에 대한 당사의 지분율은 100%입니다. 이와 함께, MZ고객의 유행 확대를 위하여, 2021년 중 냉동 PB 상품을 판매하는 ㈜루닛에 대한 추가 지분 투자를 진행하여, 상품 경쟁력을 강화하고 있습니다.

...

<p>이 기업은 소매유통업과 부동산 개발업을 주력으로 하며, 경쟁력 있는 상품과 서비스로 다양한 고객층을 확보하고 있습니다. 특히 편의점 GS25와 슈퍼마켓 GS THE FRESH, 홈쇼핑 GS SHOP를 통해 안정적인 매출을 올리고 있으며, 부동산 사업과 호텔 운영 등에 수익구조를 다변화하고 있습니다. 계열은행들은 이 기업의 MZ 고객층을 겨냥한 혁신적인 상품 개발, 금융 전문화, 그리고 디지털 전환에 주력하고 있습니다. 또한, 해외시장 진출과 다양한 사업 확장을 목표로 하고 있어 성장 가능성이 높습니다.</p>

<ul>

<li>기업 개요

<ul><li>주요 사업: 소매유통업, 부동산 개발업</li>

<li>주요 브랜드: GS25, GS THE FRESH, GS SHOP</li>

<li>부동산 관련: 개발 기획, 인테리어, 자산관리 및 매각</li>

<li>부산물 사업: 파르나스호텔 운영 및 불우 전문회사 설립</li></ul>

<li>소매유통업

<ul><li>편의점 (GS25): 24시간 영업, 상품 소매점</li>

<li>슈퍼마켓 (GS THE FRESH): 식료품 및 생활필수품 판매</li>

<li>홈쇼핑 (GS SHOP): TV쇼핑과 온라인 쇼핑</li></ul>

<li>부동산 개발사업

<ul><li>부동산 사업: 상업지선 건축 후 매각</li>

<li>개발 기획 및 운영 전문성 보유</li></ul>

<li>호텔업

<ul><li>파르나스호텔 운영: 5성급 호텔 및 비즈니스 호텔</li>

<li>파르나스 타워: 객실타워로 강남 랜드마크 구축</li></ul>

<li>불우업

<ul><li>파르나스호텔축산물: 냉동 전문 기업 목표</li>

<li>39% 지분 및 법인다각화 추진</li></ul>

<li>부동산투자사업

<ul><li>이자소득415억원(부동산투자사업) 실적</li>

<li>부동산매출 추로 투자하여 수익 확대</li></ul>

<li>기타 사업

<ul><li>㈜파르나스마트: F/F 상품 제조 및 공급</li>

<li>㈜파르나스마트: 반려동물 사업 강화</li></ul>

<li>PT. GS Retail Indonesia: 인도네시아 슈퍼마켓 운영</li>

<li>㈜루닛: MZ 고객 냉동 상품 판매</li></ul>

원문 데이터 (총 7,787자)

요약문 데이터 (태그값 포함 1,107자)

평가지표	설명	결과
지침 준수 여부	주어진 프롬프트의 조건을 얼마나 잘 따랐는지 평가	매우 우수
형식 및 구조 준수	요구된 HTML 형식과 구조를 정확히 따랐는지 평가	우수
내용 충실도	요약본이 원본의 중요한 정보를 얼마나 잘 포함하고 있는지 평가	매우 우수
언어 유창성 및 정확성	문법적 오류나 어색한 표현 없이 자연스럽게 작성되었는지 평가	매우 우수
BERTScore	요약본과 원본 간의 의미적 유사성을 측정 (0~1 사이의 값)	약 0.88

## (2) AI 모의 면접 (Domain Name : AI-Interview)

전처리 한 데이터를 통해 [질문 생성 모델] , [면접 채점 모델] 을 생성하고자 함

### (1) 모델 세팅

#### 1. 사용한 Backbone 모델

- polyglot-ko-1.3b (변경될 수도 있음)




EleutherAI : 비영리 인공지능 연구 그룹

#### 2. 프로젝트 진행 환경

- colab pro +
  - LLM 모델링을 진행하기에 자원이 부족해 한계가 있었음
  - 이에 주어진 자원 내에 올릴 수 있는 최대 모델인 1.3B 활용하고자 함


GitHub - EleutherAI/polyglot: Polyglot: Large Language Mod...

Polyglot: Large Language Models of Well-balanced Competence in Multi-languages - EleutherAI/polyglot

 <https://github.com/EleutherAI/polyglot>

EleutherAI/polyglot

Polyglot: Large Language Models of Well-balanced Competence in Multi-languages



12

Contributors

8


Issues

472

Stars

39

Forks

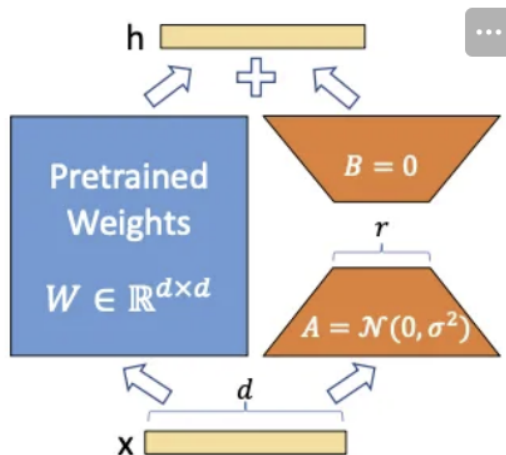


### 3. 모델링 방법

- LoRA Fine-tuning

#### 정의

- PEFT(Parameter Efficient Fine-Tuning)의 기법 중 하나
- PEFT : 전체 모델을 재학습하는 것이 아니라, Fine Tuning을 하되 Parameter Efficient 하게 해보자



- 기존 Pre-trained Model(backbone 모델)의 weight는 freeze시키고 이 weight보다 작은 새로운 weight 즉, 어댑터만 학습하는 것
- Pre-trained model의 weight는 고정된 채로, 몇 개의 dense layer만 학습시켜 downstream task의 연산량을 줄이는 효과
- 인퍼런스 할 때는 기존 파라미터 + 어댑터의 합으로 모델 사용

```
model = AutoModelForCausalLM.from_pretrained(  
    pretrained_model_name_or_path=self.config['pretrained_model_name_or_path'],  
    trust_remote_code=self.config['trust_remote_code'],  
    cache_dir=self.cacheDir,  
    local_files_only=self.config['local_files_only'])  
  
tokenizer = (AutoTokenizer.  
    from_pretrained(pretrained_model_name_or_path=self.config['pretrained_model_name_or_path'],  
                    trust_remote_code=self.config['trust_remote_code'],  
                    cache_dir=self.cacheDir,  
                    local_files_only=self.config['local_files_only'],  
                    padding_side=self.config['padding_side']))
```

pre-trained model에

```
loraAdapterScoreName = "polyglot-ko-1.3b/score"  
loraAdapterScorePath = os.path.join("models", loraAdapterScoreName, "r512-epoch100")  
scoreModel = PeftModel.from_pretrained(model, loraAdapterScorePath)  
scoreModel = scoreModel.merge_and_unload()
```

로라 어댑터를 붙여서 합(merge\_and\_load())한 모델을 사용

- 해당 방법을 통해 모델 학습 시간 및 비용을 절감할 수 있음

## (2) 사용한 평가지표

### 1. Accuracy

- 문장의 띄어쓰기를 기준으로 토큰을 나눠서 토큰 1개, 토큰 2개, 토큰 3개 각각 일치율(unigram, bigram, trigram) 비교하여 정확도 산출

```
reference_sentence = "This is example sentence."  
generated_sentence = "This is generated sentence."  
  
# 1 token  
unigram_accuracy = 1.0  
  
# 2 tokens  
bigram_accuracy = 1.0  
  
# 3 tokens  
trigram_accuracy = 0.67
```

### 2. BLEU score

- 정답이 되는 reference에 대해 모델이 생성한 문장이 얼마나 유사한 지를 평가하기 위해 사용



#### BLEU 선정 이유 부가 설명 (그냥 이해용)

- 딥러닝 분야에서 Generated Sentence task를 수행하는 평가 지표로 대표적으로 ROUGE와 BLEU를 활용함
- 해당 평가 지표의 대략적인 흐름은 source(input)에 대한 실제 정답 (reference 문장) 과 실제로 생성된 문장(output)을 비교하는 것임

**ROUGE** :  $\#\{w_{ref} \in S_{gen} | w_{ref} \in S_{ref}\} / |S_{ref}|$

**BLEU** :  $\#\{w_{gen} \in S_{ref} | w_{gen} \in S_{gen}\} / |S_{gen}|$

#### (1) ROUGE score

- Reference Sentence의 단어가 Generated Sentence에 포함되는 정도
  - 주로 abstract(요약) task 모델의 성능을 평가할 때 사용함
  - 생성된 문장이 레퍼런스 문장의 중요 키워드를 얼마나 포함하는지 측정
  - n-gram 중첩을 기반으로 계산 (단어 일치도)

#### (2) BLEU score

- Generated Sentence의 단어가 Reference Sentence에 포함되는 정도
  - 주로 기계 번역 Task 모델의 성능을 평가할 때 사용
  - n-gram 정밀도(precision)를 기반으로 계산
  - 생성된 문장이 레퍼런스 문장과 얼마나 유사한지 비교

⇒ 레퍼런스 문장과 유사한 정도를 파악하는 것이므로 BLEU score를 활용하는 것이 마땅하다고 판단

### (3) BLEU score의 계산 원리

- Generated Sentence에서 1-gram부터 4-gram까지의 n-gram이 Reference Sentence에 얼마나 포함되는지 계산 (위 수식 참고)
- 간결성 페널티(Brevity Penalty) 적용: generated Sentence가 reference sentence 보다 짧을 경우 페널티를 부과
- $BLEU = N\text{-gram 정밀도의 기하평균} * \text{간결성 페널티}$

[참고] [위키독스](#) 08. Rouge, BLEU, METEOR, SemScore 기반 휴리스틱 평가

## (3) 하이퍼파라미터튜닝

| 튜닝한 하이퍼 파라미터

### (1) epoch

- epoch를 25, 50, 75, 100번 진행했을 때의 성능을 비교함

### (2) LoRA r size

- 어댑터의 차원을 r이라고 함
- r 사이즈를 128, 256, 512 로 조정했을 때의 성능을 비교함

### (3) data size

- 데이터 사이즈를 100개, 1000개, 10000개로 조정했을 때의 성능을 비교함

### 결과?

- 질문 생성 모델의 경우 r 사이즈가 커질 수록 모델의 성능이 좋았으며, 채점 모델의 경우 r 128보다 클 경우 좋았으나 성능 차이가 미비했음.
- 채점 모델은 생성해야 할 문장의 길이가 질문 생성보다 길기 때문에 조금 더 많은 학습이 필요할 듯
- 두 모델 모두 예폭 사이즈가 커질수록 확실히 성능이 올라감

- 
- ▶ 질문 생성 모델 → 아직 최종 모델이 아니고, 작은 모델로 테스트한 결과입니다
  - ▶ 면접 채점 모델 → 아직 최종 모델이 아니고, 작은 모델로 테스트한 결과입니다