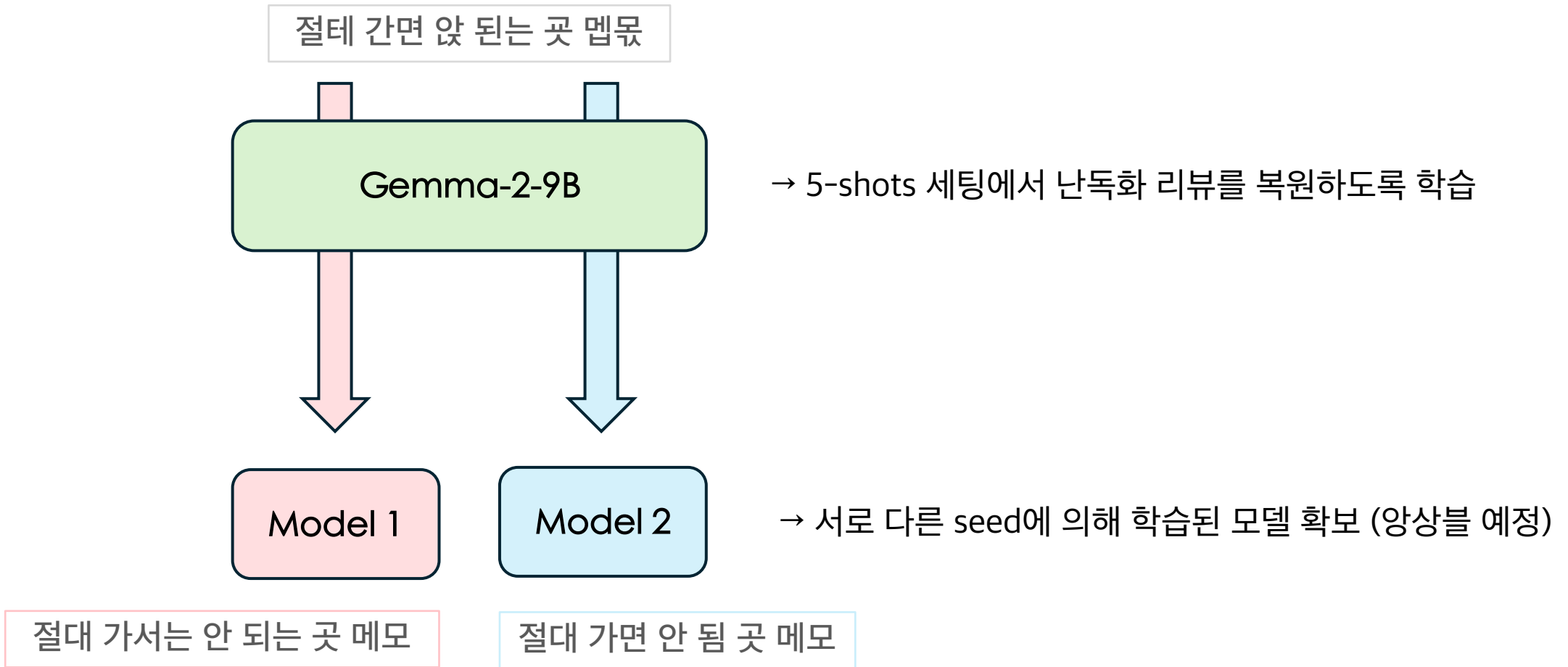


# < 난독화된 한글 리뷰 복원 AI 경진대회 > Private 7위 Solution 💡

Team 문어고앵  
정영주, 최용빈

# Overview - Training





# Overview - Inference

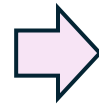
너문넓뭍 만족스러운 호템뤼에오. 푸싸눠 읍면  
꼭 चुप्헌학꼬 싶은 곳싣웨오.



Model 1

→ Model1을 활용해 1차 추론

너무너무 만족스러운 호템. 부산에 오면  
꼭 추천하고 싶은 곳.



- 너무너무 만족스러운 호템뤼에오.  
- 부산에 오면 꼭 추천하고 싶은 곳싣웨오.

→ 단어 수 매칭을 위한 후처리 및 문장 단위 분리



Model 2

Model2를 활용해 문장 단위로 2차 추론 ←  
문장 단위의 예측을 통합하여 최종 결과 도출 ←

너무너무 만족스러운 호텔이에오. 부산에 오면  
꼭 추천하고 싶은 곳이에요.



# 1. Training - Description

## 난독화된 한글 리뷰 복원 AI 경진대회

알고리즘 | 월간 데이콘 | NLP | 생성형 AI | LLM | F1 Score

🏆 상금 : 데이스쿨 프로 구독권

🕒 2025.01.06 ~ 2025.02.28 09:59 [+ Google Calendar](#)

👤 665명 📅 마감

<input type="checkbox"/>	ID	input	output
1	TRAIN_00000	별 한 게토 앓갑땀. 왜 싸람들릭...	별 한 개도 아깝다. 왜 사람들이...
2	TRAIN_00001	잠 많 작고 갑 태 좋네욤. 차 못동...	잠만 자고 갈 때 좋네요. 잠옷도...
3	TRAIN_00002	절테 간면 앓 되는 곳 맵뽕	절대 가면 안 되는 곳 메모
4	TRAIN_00003	야... 카컱 좋교 부도 뽕 뽕러쌌 ...	아... 가격 좋고 뷰도 뽕 뽕려서 ...

- **Task:** 난독화된 한글 리뷰를 원래의 명확한 내용의 리뷰로 복원하는 알고리즘 개발
- **Data:** 난독화된 한글 리뷰 (input column) 와 복원된 한글 리뷰 (output column) paired data
- **Backbone Model:** rtzr/ko-gemma-2-9b-it (<https://huggingface.co/rtzr/ko-gemma-2-9b-it>)
- **Model Task Type:** CAUSAL\_LM
- **Training Strategy:** LoRA Fine-tuning
  - ✓ Target Modules: ["v\_proj", "q\_proj", "k\_proj", "gate\_proj", "down\_proj", "o\_proj", "up\_proj"]
  - ✓ LoRA Alpha: 16, LoRA Dropout: 0.05, LoRA Rank: 32



# 1. Training - Prompting

## Instruction

"<bos>Your task is to transform the given obfuscated Korean review into a clear, correct, and natural-sounding Korean review that reflects its original meaning. Spacing and word length in the output must be restored to the same as in the input. Do not provide any description. Print only in Korean."

## Random Examples (5-shots)

"<start\_of\_turn>user  
Transform the following obfuscated review: {난독화 문장 예제}<end\_of\_turn>  
<start\_of\_turn>model  
{복원 문장 예제}<end\_of\_turn>"

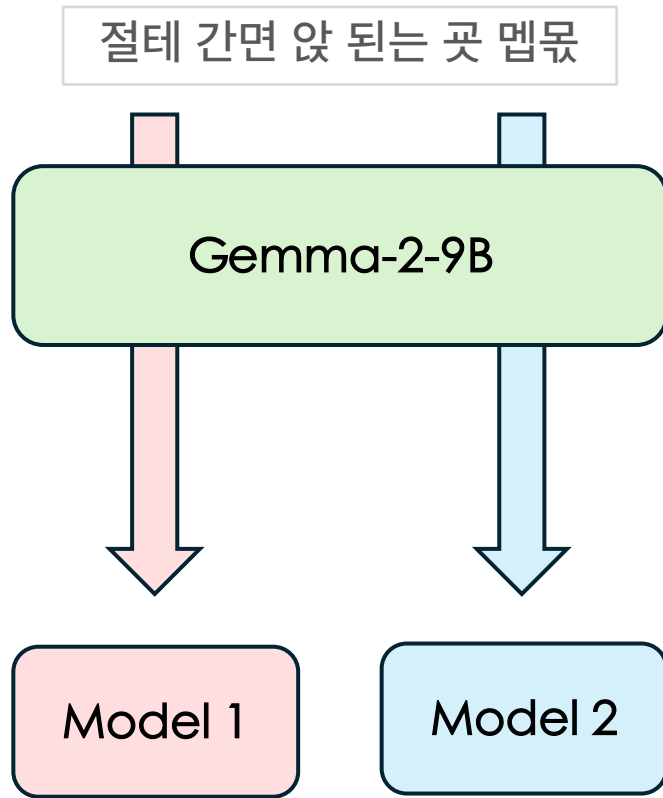
X 5

## Training Sample

"<start\_of\_turn>user  
Transform the following obfuscated review: {학습시킬 난독화 문장}<end\_of\_turn>  
<start\_of\_turn>model  
{학습시킬 복원 문장}<end\_of\_turn>"



# 1. Training - Process



- Random examples 5개를 참고하여 난독화된 한국어 문장을 복원하도록 학습
- LoRA Fine-tuning을 통해 모델 파라미터의 일부만 효율적으로 미세조정
- 서로 다른 seed setting에서 학습된 모델들 확보 (이후 앙상블 예정)



# 1. Training - Hyperparameters (with DeepSpeed)

```
torchrun --nproc_per_node 1 sft.py
--model_name_or_path rtzr/ko-gemma-2-9b-it
--torch_dtype float16
--max_seq_length 1024
--learning_rate 3e-4
--num_train_epochs 5
--per_device_train_batch_size 1
--gradient_accumulation_steps 8
--logging_steps 10
--save_strategy epoch
--attn_implementation eager
--warmup_ratio 0.1
--ddp_find_unused_parameters False
--gradient_checkpointing
--deepspeed ../stage1.json → https://github.com/whybe-choi/dacon-korean-review-deobfuscation/blob/main/stage1.json
--fp16
```



## 2. Inference - Prompting

### Instruction

"<bos>Your task is to transform the given obfuscated Korean review into a clear, correct, and natural-sounding Korean review that reflects its original meaning. Spacing and word length in the output must be restored to the same as in the input. Do not provide any description. Print only in Korean."

### Random Examples (n-shots)

"<start\_of\_turn>user  
Transform the following obfuscated review: {난독화 문장 예제}<end\_of\_turn>  
<start\_of\_turn>model  
{복원 문장 예제}<end\_of\_turn>"

x n

### Target Sample

"<start\_of\_turn>user  
Transform the following obfuscated review: {복원할 테스트 난독화 문장}<end\_of\_turn>  
<start\_of\_turn>model  
→ 이후 부분을 모델이 추론"





## 2. Inference - The first inference

너문넓뭇 만족스러운 효템뤼에오. 푸싸눠 읍면  
꼭 चुप्학꼬 싶은 곳싰뤼에오.



Model 1

너무너무 만족스러운 효템. 부산에 오면  
꼭 추천하고 싶은 곳.

- Model1을 활용하여 첫번째 추론 수행
- 문장 길이 문제로 shot 개수를 4개로 조정
- Sampling없이  
num\_beams=1, max\_new\_tokens=1024로 설정



## 2. Inference - Issue

너무너무 만족스러운 호텔이에요. 푸사네 읍면 꼭 추천하고 싶은 곳신워요.



Model 1

오답처리

추론 결과 너무너무 만족스러운 호텔. 부산에 오면 꼭 추천하고 싶은 곳.

예상 답안 너무너무 만족스러운 호텔이에요. 부산에 오면 꼭 추천하고 싶은 곳이에요.



제출 답안과 정답이 정확한 위치에서 일치해야 하므로,  
중간 위치에서 추론이 잘못될 경우 이후 문장들이 올바르게 추론된다고 하더라도 좋은 평가를 받기가 어려움.



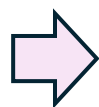
## 2. Inference - Post processing

- Training data를 활용해 형태소 기반 단어 사전 구축. → Test data의 난독화 문장을 단어 사전 기반으로 대체하여 처리  
e.g.) 아깍: {'약깍', '악깍', ...}
- 입력 (input column) 과 추론 결과 간 단어 길이 매칭 작업 수행
  - ✓ 입력 단어보다 추론된 단어가 긴 경우: 입력 문장의 길이에 맞추어 잘라내기
  - ✓ 입력 단어보다 추론된 단어가 짧은 경우: 단어 사전 기반으로 처리된 난독화 리뷰를 남은 길이만큼 이어 붙이기

입력 문장	너문넓뭍 만족스러운 호템뤼에오. 푸싸눠 읏면 콧 츠쩐학꼬 싶은 콧쑤웨오.
Model1 추론 결과	너무너무 만족스러운 호템. <u>부산에 오면 꼭 추천하고 싶은 곳.</u> <span style="color: red;">→ 오답처리</span>
후처리 결과	너무너무 만족스러운 호템뤼에오. 부산에 오면 꼭 추천하고 싶은 곳 <u>쑤웨오.</u> <span style="color: blue;">← 입력 문장과 단어 길이 맞추기!</span>

- 후처리 완료 후 문장 단위로 분리

너무너무 만족스러운 호템. 부산에 오면  
꼭 추천하고 싶은 곳.



- 너무너무 만족스러운 호템뤼에오.
- 부산에 오면 꼭 추천하고 싶은 곳쑤웨오.



## 2. Inference - The second inference

- 너무너무 만족스러운 호텔뤼에오.
- 부산에 오면 꼭 추천하고 싶은 곳쥬웨오.



Model 2

너무너무 만족스러운 호텔이에오. 부산에 오면  
꼭 추천하고 싶은 곳이에오.

- Model2을 활용하여 두번째 추론 수행
- 첫번째 추론과 달리 한 문장씩 추론  
→ 문장 길이가 짧아지므로 shot을 5개로 조정
- Sampling없이  
num\_beams=5, max\_new\_tokens=1024로 설정



## 2. Inference - Final post processing

- 너무너무 만족스러운 호텔이에요.  
- 부산에 오면 꼭 추천하고 싶은 곳이에요.



너무너무 만족스러운 호텔이에요. 부산에 오면  
꼭 추천하고 싶은 곳이에요.

- 문장 단위로 복원된 리뷰를 기존 난독화 리뷰 문단의 형태로 다시 통합하는 과정
- 문장 단위로 리뷰를 복원하였음에도 글자 수가 일치하지 않는 부분에 대해서는 이전 post processing에서와 같이 글자수 매칭 작업 다시 수행
- 최종 복원 완료



# Appendix - Environment

## Computing Resources

Runpod A100 PCIe, A6000

## Requirements

Python==3.10

torch==2.4.0

peft==0.11.1

trl==0.9.4

transformers==4.43.3

datasets==2.20.0

accelerate==0.29.3

bitsandbytes==0.42.0

wandb==0.17.4

deepspeed==0.15.4

hgtk==0.2.1

kss==6.0.4

kiwipiepy==0.20.3

python-mecab-ko==1.3.7

vllm==0.5.4

**Available Code:** <https://github.com/whybe-choi/dacon-korean-review-deobfuscation>