NLP Final Project

GPT-2 Fine-tuning – Sentimental Analysis / Paraphrase Detection / Sonnet Generation

Presented by: Jaewon Lee, Byeongmin Kang
2025.06.18

Contents

Introduction

GPT-2 Fine tuning

- **Sentimental Analysis** - SST, CFIMDB corpus dataset / Classify task
- **Paraphrase Detection** - QQP dataset / cloze-style task (yes/no generation)
- **Sonnet Generation** - Shakespeare Sonnet dataset / multi-token generation task

Experiments Setting

- 동국대학교 서버 - **GPU: A6000 48gb x 1**

- Train: quora_train.csv / Validation, Test: quora_dev.csv

- Model Size: GPT-2
- Batch Size: 96
- Learning Rate: default 2e-5 / diff with experiments.
- Seed: 11711

- about 15-30 min. per epoch.

Baseline

- Full Finetuning.
- R-Drop (KL) / lr scheduler

Method	Accuracy	F1 Score	others
Baseline(_250523_default_paraphrase)	0.9081	0.9019	epoch=10, lr=1e-5
Baseline(_250614_pooling, last)	0.8919	0.8849	epoch=20, lr=2e-5
Baseline(_250613_default_BackTranslation)	0.8854	0.8781	epoch=10, lr=2e-5

SimCSE

- **SimCSE**: Simple Contrastive Learning of Sentence Embeddings (Gao et al., 2021)
- two forward pass - dropout mask x 2
- InfoNCE loss

$$\ell_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'})/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'})/\tau}}$$

SimCSE(_250522_SimCSE)	0.8946	0.8879	epoch=3, lr=2e-5
SimCSE(_250523_simcse_etc)	0.8682	0.8594	epoch=3, lr=2e-5

Adversarial

- Improving Paraphrase Detection with the **Adversarial** Paraphrasing Task (Animesh Nighojkar, John Licato., 2021)

- APT dataset generation - low BLEURT score (like hard negative)

Adverserial(_250611_Adverserial_gem)	0.8909	0.8836	epoch=30, lr=2e-5

Multitask

- End-to-End **Multi-Task** Learning with Attention (Liu et al., 2019)
- multi task end-to-end train -> task별 weighted sum loss.

- Just re-finetuning - ETPC dataset

Multitask(_250615_multitask2)	0.7709	0.7484	epoch=10/20, lr=2e-5

PeFT

- **LoRA** (Hu et al., 2022), **DoRA** (Liu et al., 2024)

PeFT(_250614_PeFT, LoRA)	0.8707	0.8623	epoch=20, lr=5e-5 r=16, alpha=32, dropout=0.1
PeFT(_250614_PeFT, LoRA)	0.8599	0.8498	epoch=20, lr=5e-5 r=8, alpha=16, dropout=0.1
PeFT(_250614_PeFT, DoRA)	0.8702	0.8616	epoch=20, lr=5e-5 r=16, alpha=32, dropout=0.1
PeFT(_250614_PeFT, DoRA)	0.8601	0.8501	epoch=20, lr=5e-5 r=8, alpha=16, dropout=0.1

Pooling

- last -> mean pooling

Pooling(_250614_pooling, mean)	0.8872	0.88	epoch=20, lr=2e-5

BackTranslation

- BackTranslation (Corbeil and Ghadivel, 2020)

BET: A BACKTRANSLATION APPROACH FOR EASY DATA AUGMENTATION IN TRANSFORMER-BASED PARAPHRASE IDENTIFICATION CONTEXT

- ar, de, es, fr, ru, zh Backtranslate -> data augment

BackTranslation(_250612_BackTranslation)	0.8784	0.871	epoch=20, lr=2e-5

Results

Method	Accuracy	F1 Score	others
Baseline(_250523_default_paraphrase)	0.9081	0.9019	epoch=10, lr=1e-5
Baseline(_250614_pooling, last)	0.8919	0.8849	epoch=20, lr=2e-5
Baseline(_250613_default_BackTranslation)	0.8854	0.8781	epoch=10, lr=2e-5
SimCSE(_250522_SimCSE)	0.8946	0.8879	epoch=3, lr=2e-5
SimCSE(_250523_simcse_etc)	0.8682	0.8594	epoch=3, lr=2e-5
Adverserial(_250611_Adverserial_gem)	0.8909	0.8836	epoch=30, lr=2e-5
Multitask(_250612_Multitask)	0.8891	0.8505	epoch=20, lr=2e-5
Multitask(_250615_multitask2)	0.7709	0.7484	epoch=10/20, lr=2e-5
PeFT(_250614_PeFT, LoRA)	0.8707	0.8623	epoch=20, lr=5e-5 r=16, alpha=32, dropout=0.1
PeFT(_250614_PeFT, LoRA)	0.8599	0.8498	epoch=20, lr=5e-5 r=8, alpha=16, dropout=0.1
PeFT(_250614_PeFT, DoRA)	0.8702	0.8616	epoch=20, lr=5e-5 r=16, alpha=32, dropout=0.1
PeFT(_250614_PeFT, DoRA)	0.8601	0.8501	epoch=20, lr=5e-5 r=8, alpha=16, dropout=0.1
Pooling(_250614_pooling, mean)	0.8872	0.88	epoch=20, lr=2e-5
BackTranslation(_250612_BackTranslation)	0.8784	0.871	epoch=20, lr=2e-5

Experiments Setting

동국대학교 서버 사용(cs.dongguk.edu 102번 포트 linuxserver2)

- GPU: NVIDIA A6000 48GB x 1 사용

```
flareon@gangbyeongmins-MacBook-Pro ~ % ssh -p 102 2020112534@cs.dongguk.edu
Last login: Mon Jun 16 04:31:02 2025 from 203.251.170.194
Error changing group of pid 2083446: Cgroup does not exist
2020112534@linuxserver2:~$ nvidia-smi
Tue Jun 17 14:31:46 2025
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.230.02             Driver Version: 535.230.02   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA RTX A6000               Off | 00000000:01:00.0 Off |                  Off |
| 30%   36C    P8              26W / 300W |     10MiB / 49140MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
|   1  NVIDIA RTX A6000               Off | 00000000:25:00.0 Off |                  Off |
| 30%   36C    P8              23W / 300W |     10MiB / 49140MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
|   2  NVIDIA RTX A6000               Off | 00000000:41:00.0 Off |                  Off |
| 30%   35C    P8              19W / 300W |     10MiB / 49140MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
|   3  NVIDIA RTX A6000               Off | 00000000:81:00.0 Off |                  Off |
| 30%   39C    P8              35W / 300W |     10MiB / 49140MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
|   4  NVIDIA RTX A6000               Off | 00000000:A1:00.0 Off |                  Off |
| 30%   35C    P8              23W / 300W |     10MiB / 49140MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
|   5  NVIDIA RTX A6000               Off | 00000000:C1:00.0 Off |                  Off |
| 56%   81C    P2             277W / 300W |  46765MiB / 49140MiB |    100%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
|   6  NVIDIA RTX A6000               Off | 00000000:E1:00.0 Off |                  Off |
| 30%   35C    P8              25W / 300W |     10MiB / 49140MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A      4208      G   /usr/lib/xorg/Xorg                            4MiB |
|    1   N/A  N/A      4208      G   /usr/lib/xorg/Xorg                            4MiB |
|    2   N/A  N/A      4208      G   /usr/lib/xorg/Xorg                            4MiB |
|    3   N/A  N/A      4208      G   /usr/lib/xorg/Xorg                            4MiB |
|    4   N/A  N/A      4208      G   /usr/lib/xorg/Xorg                            4MiB |
|    5   N/A  N/A      4208      G   /usr/lib/xorg/Xorg                            4MiB |
|    5   N/A  N/A   2037265      C   python                                    46726MiB |
|    6   N/A  N/A      4208      G   /usr/lib/xorg/Xorg                            4MiB |
+---------------------------------------------------------------------------------------+
```

Baseline

- sonnet_generation_base.py
- **top-p sampling**을 이용한 generate() 함수를 통해 기본적인 소네트 생성을 수행

Prefix-Tuning

- **sonnet_generation_Prefix.py**

- 입력 앞에 짧은 힌트를 붙여서, 모델이 원하는 스타일이나 태스크에 맞게 반응하도록 유도하는 방법

- 기존 모델 전체를 학습시키는 대신 → 작은 prefix만 학습해서 학습 속도와 파라미터 수를 줄이기 위해
- 기존 모델은 그대로 두고 → prefix만 바꿔서 다양한 태스크에 쉽게 적응 가능

Contrastive Search

- **sonnet_generation_CS.py**

- Li et al. (2022)의 Contrastive Decoding 방식을 기반으로 구현.
- top-k 후보군 중에서 hidden state 간 유사도(repulsion score)를 계산 → 이를 로짓 점수에서 감산하여 대표성이 높은 토큰 선택

옵션명	역할	기대 효과
dynamic_alpha	생성 step이 늘어날수록 α값(유사도 감점 강도)을 선형 증가	초반에는 유연하게, 후반에는 더 확실하게 억제 → 문장 후반부 안정화
use_mean_sim	후보 토큰과 **과거 히든 상태 전체 평균 유사도** 사용	전체 맥락에서 다양성 확보 (vs max는 가장 비슷한 하나만 반영)
use_repetition_penalty	이미 등장한 토큰의 로짓 점수를 나눠서 반복 억제	동일 단어나 문장 반복을 방지 → 표현 다양성 증가

Beam Search

- **sonnet_generation_BS.py**

- 전통적인 Beam Search 알고리즘을 기반으로 구현
- top-k 토큰 후보를 탐색하여 시퀀스를 확장
- 길이 정규화(length_penalty)와 n-gram 반복 차단 기능을 포함

MBR

- **sonnet_generation_MBR.py**

- Minimum Bayes Risk (MBR) reranking 기법을 적용
- Beam Search를 통해 생성된 상위 n-best 후보들 간 pairwise CHRF 유사도

-> 평균 유사도가 가장 높은 후보를 최종 출력으로 선택

Candidate Ensemble + MBR

- **sonnet_generation_CE.py**

- Top-p Sampling, Contrastive Search, Beam Search를 통해 생성된 다양한 후보군을 모두 통합
- 이들 간의 pairwise CHRF 유사도를 기반으로 MBR reranking을 수행

Results

Method	Accuracy	others
Baseline (Top-p Sampling)	39.7379	기본 generate 함수, top-p=0.9, temp=1.2
Prefix-Tuning	35.0762	연속 prefix embedding
Contrastive Search (all False)	35.6912	기본 contrastive 설정 (3옵션 모두 꺼짐)
Contrastive Search (only dynamic_alpha)	36.3526	dynamic_alpha=True
Contrastive Search (only use_mean_sim)	35.6912	use_mean_sim=True
Contrastive Search (only repetition_penalty)	24.8650	use_repetition_penalty=True
Contrastive Search (dynamic+mean_sim)	37.7542	best setting (두 옵션만 True)
Beam Search	33.4421	beam_size=5, length_penalty=0.6
MBR	37.7483	Beam 후보 기반 reranking
Candidate Ensemble + MBR	37.6524	Top-p, Contrastive, Beam 통합후 rerank

Fine-Tuning

- 가장 안정적이고 높은 성능을 보였던 baseline (Top-p Sampling) 방식을 기반으로 전체 GPT-2 모델 파라미터를 fine-tuning하였다.

- 학습 시 forward()는 hidden state 전체를 활용해 logit을 생성하였으며, Top-p 샘플링을 통한 generation 방식은 그대로 유지하였다.

학습 구조는 기존 train() 함수를 개선하여 다음과 같은 기법을 새롭게 추가하였다.

- get_linear_schedule_with_warmup()을 통한 학습률 스케줄링 적용
- clip_grad_norm_()을 통한 그래디언트 클리핑 도입 (max norm=1.0)
- 매 epoch마다 Dev 세트 CHRF 평가 및 최고 성능 상위 3개 모델만 저장
- 불필요한 체크포인트 파일은 자동 삭제하여 저장 공간 최적화

Fine-Tuning

Batch Size	Learning Rate	Epochs	CHRF Score	Notes
32	1e-6	1000	41.8519	Saved at epoch 594
32	5e-5	1000	42.6414	Saved at epoch 469
32	1e-4	1000	42.4880	Saved at epoch 188
8	1e-5	200	43.2194	Saved at epoch 74 (SOTA)

Paraphrase Detection

- Baseline이 기본적으로 가장 괜찮은 성능
- 그러나 여러 Method에서 hyperparameter 최적화 시 Baseline보다 더 성능이 개선될 수 있음을 확인.

- hyperparameter searching을 통한 최적화를 이후 과제로 남겨둠.
- Ensemble 기법을 활용하면 더 좋은 성능을 확인할 수 있음으로 사료되나, 확인이 필요함.

- Adversarial / Multi-task 등의 방식에서 논문의 Method를 그대로 사용하지 x.
- 이에 대한 추가 실험이 필요.

Sonnet Generation

- Baseline (generate) 구조가 기본적으로 가장 안정적인 성능
  - fine-tuning을 통해 CHRF 기준 가장 높은 점수를 달성

- Prefix-Tuning, Contrastive Search, Beam Search, MBR 등 다양한 방법론 적용.
  - 일부 전략은 비슷한 수준의 성능을 보였으나 완전한 fine-tuning에는 미치지 못함.

- 추후 다양한 decoding 기법에 대한 조합 탐색(Ensemble 등) 및 prefix 기반 세분화 학습 전략에 대한 추가 실험 필요.

Thank You