

GPT-2 기반 다중 태스크 자연어 처리

: 감정 분석, 패러프레이즈 탐지, 소네트 생성

Abstract

본 연구는 GPT-2 기반의 소형 언어 모델을 활용하여 감정 분석, 패러프레이즈 탐지, 소네트 생성이라는 세 가지 자연어 처리 태스크를 통합적으로 수행하였다. 감정 분석 태스크에서는 클래스 불균형 문제를 해결하기 위한 데이터 rephrasing 기법과 joint-cascade 구조를 제안하였고, 패러프레이즈 탐지에서는 LoRA 기반 경량화 모델에 hard negative priority(HNP)를 적용하여 추가적인 성능 향상을 도모하였다. 또한, 소네트 생성 태스크에서는 전통적 운율 구조의 특성을 반영하기 위해 rhyme loss를 도입하고, 시 생성 시 14행 구조를 보장하는 후처리 기법을 함께 적용하였다. 실험 결과, 각 태스크별로 제안된 방법이 성능에 긍정적인 영향을 주지 못한 부분도 있으나, 소네트 생성 task의 rhyme accuracy 지표 등에서 기존 baseline 대비 유의미한 개선을 확인할 수 있었다. 본 연구는 제한된 자원 환경에서의 다중 NLP 태스크 수행 가능성과 추후에 추가적인 실험을 통한 task-specific 개선 접근 방향을 제시하고자 한다.

1. Introduction

최근 자연어 처리(Natural Language Processing, NLP) 분야에서는 범용 언어 모델을 기반으로 다양한 태스크를 동시에 수행하려는 시도가 활발히 이루어지고 있다. 특히 GPT 계열의 사전학습 언어 모델은 적은 양의 태스크 특화 학습 데이터만으로도 감정 분석, 질문 응답, 텍스트 생성 등 다양한 작업을 효과적으로 수행할 수 있어 주목받고 있다.

그러나 실제 응용 단계에서는 세 가지 한계점이 동시에 대두된다. 첫째, 감정 분석에서는 클래스 간 불균형으로 인해 모델이 특정 클래스에 편향되기 쉽다. 둘째, 대규모 데이터셋 기반 패러프레이즈 탐지에서는 훈련 시간이 과도하게 소요되며, 중요한 오류 샘플을 효과적으로 활용하기 어렵다. 셋째, 창의적 텍스트 생성 과제에서는 문법적 정확성뿐 아니라 시적 구조와 운율(rhyme) 같은 정성적 특성까지 고려해야 하는 추가적인 도전 과제가 존재한다.

본 프로젝트에서는 이러한 한계점들을 극복하기 위해 GPT-2 기반의 모델을 바탕으로 감정 분석, 패러프레이즈 탐지, 소네트 생성이라는 세 가지 task를 각각 해결하고자 하였다. 이를 위해 각 태스크에 맞는 핵심 기법들을 도입하였으며, 성능 개선 효과를 정량적으로 검증하였다. 본 보고서에서는 각 기법의 설계 의도, 구현 방법, 실험 결과를 중심으로 통합적 분석을 제시한다.

2. Related Work

본 연구에서는 감정 분석, 패러프레이즈 탐지, 소네트 생성이라는 세 가지 자연어 처리 태스크를 대상으로 GPT-2 모델 기반의 통합적 실험을 수행하였다. 이 장에서는 각 태스크에 대해 기존 연구 동향 및 대표적인 접근 방법을 개관하고자 한다.

먼저 감정 분석(Sentiment Classification) 분야에서는 Stanford Sentiment Treebank(SST)나 CFIMDB와 같은 감성 태깅 데이터셋을 기반으로 다양한 신경망 기반 분류 모델이 제안되어 왔다. 특히 BERT와 RoBERTa와 같은 사전학습 기반 모델은 문맥적 감정 정보를 효과적으로 포착하여 높은 성능을 보였다. 최근에는 클래스 불균형 문제 해결을 위해 데이터 증강이나 단계적 분류 구조(cascade classification) 등이 함께 활용되고 있다.

패러프레이즈 탐지(Paraphrase Detection) 분야에서는 Quora Question Pair와 Microsoft Research Paraphrase Corpus(MRPC) 등이 대표적인 벤치마크로 사용되며, Siamese network 구조와 contrastive loss 기반 모델이 널리 연구되었다. 최근에는 Transformer 계열 모델(BERT, DeBERT 등) 기반의 sentence-pair classification 접근이 표준으로 자리 잡았으며, LoRA와 같은 파라미터 효율화 기법이 대규모 모델의 학습 부담을 줄이는 데 기여하고 있다.

소네트 생성(Sonnet Generation)과 같은 창의적 텍스트 생성 분야는 일반적인 language modeling을 넘어 시 구조, 운율(rhyme), 행 수(line length) 등 정성적 제약 조건을 만족시키는 것이 핵심 과제이다. GPT-2 기반 시 생성 연구들은 보통 rhyme dictionary 또는 발음 사전(예: CMU Pronouncing Dictionary)을 기반으로 rhyme pattern을 도입하거나, rhyme constraint를 반영하는 추가적인 loss function을 설계하는 방식으로 접근한다.

본 연구는 위와 같은 기존 연구 중 일부를 바탕으로, 각 태스크에 적합한 GPT-2 기반 구조를 설계하고, 경량화, 리프레이징, rhyme loss 등의 다양한 아이디어를 성능 향상을 도모하였다.

3. Method

3.1 감정 분석 (sentiment classification)

감정 분석 태스크에서는 Stanford Sentiment Treebank(SST)와 CFIMDB 데이터셋을 활용하여 구현되었으며, 클래스 불균형 문제와 fine-grained 감정 분류 문제를 해결하기 위한 두 가지 핵심 전략이 적용되었다.

첫 번째 전략은 데이터 리프레이징(Rephrasing) 기반의 데이터 증강이다. 클래스 간 샘플 수 불균형 문제를 완화하기 위해, 소수 클래스에 속한 문장들을 다양한 표현으로 재구성하여 데이터 분포의 밸런스를 맞추었다.

두 번째 전략은 Joint-Cascade 분류 구조이다. 감정 클래스를 한 번에 5단계로 분류하기 어려운 문제를 해결하기 위해, 먼저 감정 범위를 coarse하게 두 그룹(0-2 vs 3-4)으로 분류한 뒤, 해당 결과를 기반으로 두 개의 fine classifier가 각각 세부 감정 클래스를 분류하도록 설계하였다. 각 단계는 GPT-2 기반 분류 모델을 활용하였으며, 전체 구조는 joint training이 가능하도록 설계되었다.

3.2 패러프레이즈 탐지 (Paraphrase Detection)

패러프레이즈 탐지 태스크는 Quora Question Pair 데이터셋을 기반으로 진행되었으며, 학습 시간 단축과 성능 향상을 위해 LoRA와 Hard Negative Priority (HNP) 기법이 적용되었다.

먼저, LoRA(Low-Rank Adaptation) 기법은 GPT-2 모델 일부 weight matrix에 대해 low-rank 행렬로 분해하여 학습 파라미터 수를 대폭 감소시키면서도 성능 저하 없이 빠른 학습이 가능하도록 하였다.

또한 HNP(Hard Negative Priority) 기법은 초기 5 epoch 학습 후, 모델이 잘못 분류했으나 confidence가 높은 하드 네거티브 샘플을 추출하여, 해당 샘플들에 대해 추가적인 1 epoch fine-tuning을 진행함으로써 모델의 분류 경계를 보완하였다.

3.3 소네트 생성 (Sonnet Generation)

소네트 생성 태스크는 GPT-2 기반 언어모델을 사용하여 Shakespeare 스타일의 14행 시를 생성하는 방식으로 구성되었다. 기존의 perplexity, BERTScore 외에도 Rhyme Accuracy라는 평가 지표를 새롭게 추가하였으며, 해당 지표의 성능 향상을 위해 Rhyme Loss 기법을 도입하였다.

Rhyme Loss는 각 행의 끝 단어 embedding 간의 cosine similarity를 고려하여, 인접한 행 간 운율(rhyme)이 맞도록 유도하는 보조 손실 함수이다. 또한, 생성된 시가 반드시 14행을 가지도록 보장하기 위해, 최대 5회 재생성을 시도하며, 그 이후에도 기준을 만족하지 않으면 초과 행은 자르고 부족한 경우는 빈 행으로 보완하는 후처리 로직을 추가하였다. 이러한 방식은 시의 rhyme accuracy를 높이는 데 긍정적인 영향을 주었다.

4. Experiments

4.1 실험 환경 및 설정

본 실험은 Google Colab Pro 환경에서 진행되었으며, 주요 라이브러리는 Pytorch 2.0, Huggingface Transformers, scikit-learn 등이 사용되었다. 모델은 GPT-2 base 구조를 기반으로 각 태스크에 맞게 커스터마이징하여 학습하였으며, 하드웨어는 A100 GPU를 활용하였다.

항목	감정 분류	패러프레이즈 탐지	소넷 생성
optimizer	AdamW	AdamW	AdamW
Learning Rate	1e-5	1e-5	1e-5
Batch Size	32	64	16
Epoch	30	5 + 1(HNP)	20
Early stopping (patience)	5	x	3

4.2 데이터셋

- Sentiment Classification
 - SST-5: 감정을 5단계(very negative ~ very positive)로 분류
 - CFIMDB: 이진 감정 분류 (positive / negative)
- Paraphrase Detection
 - Quora Question Pair: 문장쌍이 동의어 관계인지 여부 판단하여 'yes', 'no' 생성
- Sonnet Generation
 - Shakespeare의 소네트 원문을 기반으로 학습 (총 150편, held-out 12편 평가용)

4.3 평가 지표

- 감정 분석, 패러프레이즈 탐지: Accuracy, F1-score
- 소네트 생성: BERTScore, Perplexity, Rhyme Accuracy

4.4 실험 결과 및 비교

(1-1) Sentiment Classification (단일 모델 vs Cascade 모델)

모델 구조	SST		CFIMDB	
	Dev Accuracy	Dev F1-score	Dev Accuracy	Dev F1-score
rephrasing X (단일)	0.4614	0.4108	0.9429	0.9428
rephrasing O (단일)	0.4514	0.4403	0.9551	0.9551
Joint-Cascade (2단)	0.44142	0.42358	-	-

※ Cascade 모델은 rephrasing 적용된 데이터를 학습

※ Cascade 구조는 SST 데이터에만 적용

(Cascade 버전의 코드에서도 CFIMDB에는 단일 모델 적용)

- <rephrasing 기법의 효과>

SST 데이터셋에서는 리프레이징 적용 시 Accuracy가 소폭 감소했으나, F1-score는 0.4108 → 0.4403으로 보다 유의미한 향상을 보였다. 이는 소수 클래스에 대한 표현 다양성 증대가 모델의 균형 잡힌 분류 성능을 높였음을 시사한다. 반면 CFIMDB에서는 Accuracy와 Accuracy와 F1-score 모두 소폭 상승하였으며, 리프레이징의 긍정적인 영향을 확인할 수 있다.

- <단일 모델 vs Cascade 모델>

단일 모델과 비교했을 때 cascade는 모든 지표에서 성능이 하락함을 보인다. 이는 coarse classifier의 초기 분류 오류가 fine classifier로 전이되며 전체 성능이 낮아진 것으로 추정된다.

(1-2) Sentiment Classification (Cascade 모델_2단 vs 3단)

epoch	모델 구조	SST	
		Dev Accuracy	Dev F1-score
30	2단	0.44142	0.42358
	3단	0.41871	0.41813
50	2단	0.42325	0.40681
	3단	0.42598	0.40769

※ 3단 cascade 구조: 각 모델이 전부 이진 분류를 하기 위해서 고안된 구조

- coarse/ (0,1,2)-> 0, (3,4)-> 1
- fine 0/ (0,1)-> 0, 2-> 1 => fine 2/ 0-> 0, 1-> 1
- fine 1/ 3-> 0, 4-> 1

- <Cascade: 2단 vs 3단>

epoch 30 시점에서는 2단 Cascade 구조가 3단보다 더 높은 성능을 기록하였으나, epoch 50에서는 3단 구조가 오히려 더 높은 성능을 가진다. 이는 cascade 구조의 깊이가 깊어질수록 표현력은 증가하지만, 학습 안전성은 감소하는 구조적 특성을 반영한다. 따라서 3단 모델은 충분한 학습이 진행될 경우 fine-grained 감정 분류에서 더 높은 성능을 달성할 가능성이 있으며, 이를 위해 더 많은 epoch가 요구될 수 있다.

- <Epoch: 30 vs 50>

Epoch를 30에서 50으로 증가시킨 경우, 3단 Cascade 모델의 Accuracy는 소폭 상승하였으나, 나머지 성능 지표들은 모두 하락하였다. 특히 F1-score의 하락은 모델이 특정 다수 클래스에 과적합되면서 전체적인 분류 균형이 무너졌을 가능성을 시사한다. 또한 3단 구조는 학습이 진행될수록 fine classifier 간 수렴 속도 차이나 오류 누적이 심화될 수 있으며, 이로 인해 분류 경로 간 편향이 발생할 수 있다. 이러한 결과는 fine-grained task에서 Accuracy와 같은 단일 지표에 의존할 경우, 모델의 실제 성능을 과대평가할 수 있음을 보여준다.

=> 3단 Cascade 모델은 이론적으로 더 높은 표현력을 바탕으로 향후 fine-grained 감정 분류에서의 성능 향상이 기대된다. 그러나 본 프로젝트에서는 제한된 학습 자원과 시간, 그리고 실제 구현에서의 안정성을 고려할 때, 성능과 신뢰도 모두에서 더 일관된 결과를 보인 2단 모델과, 30 epoch 기준에서 가장 높은 성능을 기록한 단일 base모델 (rephrasing 적용)을 최종 결과물로 선정하였다.

(2) Paraphrase Detection

모델	Dev Accuracy	F1-score
GPT-2 (기본 5 epoch)	0.8288	0.8217
+ HNP (1 epoch)	0.8050	0.8015
+ HNP (+ train데이터)	0.8189	0.8133

- <Hard Negative Priority의 효과>

GPT-2 모델을 5 epoch 동안 학습하여 얻은 best 모델의 성능을 baseline으로 설정하였다. 이후 해당 모델에 HNP 기법을 적용하여 추가 1 epoch fine-tuning을 수행한 결과, 오히려 성능이 하락하는 현상이 관찰되었다.

이는 HNP에 사용하기로 선별된 hard negative sample들이 학습 과정에서 모델을 과도하게 특정 오류 분포에 집중시키며, 전체적인 일반화 성능을 저해했을 가능성을 시사한다. 특히 confidence 기반으로 선별된 오류 샘플은 예측 확률은 높지만 실제 정답과는 불일치하므로, fine-tuning 시 노이즈로 작용했을 가능성이 존재한다.

이러한 문제를 완화하고자 Hard Negative sample에 원래 train 데이터 일부를 병합하여 학습을 진행한 결과, 이전보다 성능이 소폭 향상되었으나 여전히 baseline 모델의 성능에는 미치지 못했다. 이는 향후 HNP 기법의 효과를 극대화하기 위해 confidence threshold의 정밀한 조정이나, label filtering 등 복합적인 추가 보완 전략이 필요함으로 해석된다.

=> HNP 성능 개선을 위한 여러 전략이 존재하나 본 프로젝트에서는 제한된 학습 자원과 시간을 고려하여 기본 5 epoch에 학습으로부터 나온 best 모델에 hard negative sample + train 데이터 일부를 1 epoch 추가학습 시킨 모델을 최종 결과물로 선정하였다.

(3-1) Sonnet Generation_정량적 평가

구성	BERTScore	Perplexity	Rhyme Accuracy
기본	0.789	35.68	0.00
+ Rhyme Loss	0.785	34.86	0.05
+ 후처리 (14줄)	0.785	34.15	0.12

Sonnet Generation task에서는 BERTScore, Perplexity, Rhyme Accuracy 세 가지 지표를 통해 모델의 정량적 성능을 평가하였다.

우선, 기본 모델은 Perplexity 35.68, BERTScore 0.789, Rhyme Accuracy 0.00을 기록하였다. 이는 rhyme 구조를 전혀 고려하지 않은 상태에서 문법적

자연스러움은 확보되었으나, 운율 구조는 완전히 무시된 결과임을 보여준다.

다음으로, rhyme 구조를 반영한 Rhyme Loss를 적용한 결과, Perplexity는 34.86로 감소하고 Rhyme Accuracy는 0.05로 상승하였다. 일반적으로 rhyme 구조 강화를 위한 학습은 운율 패턴에 집중하게 만들어 의미적 일관성이나 문법적 자연스러움 측면에서 악영향을 줄 수 있다. 그러나 본 모델에서는 BERTScore의 큰 손실 없이 의미적 품질을 유지하였고, Perplexity 또한 동반 개선되는 결과를 보였다. 이는 rhyme 구조 반영과 문장 전체 품질 간의 균형을 성공적으로 달성했음을 시사한다.

마지막으로 rhyme loss 기반 모델에 14줄 구조 보장 후처리(postprocessing)를 추가한 결과, Rhyme Accuracy는 0.12로 가장 높아졌고, Perplexity 역시 34.15로 가장 낮은 값을 기록하였다. BERTScore는 0.785로 유지되며, 의미 손실 없이 운율적·형식적 완성도를 높일 수 있음을 확인하였다.

(3-2) Sonnet Generation_정성적 평가

정량적 지표 외에도 생성된 소넛의 문학적 품질과 문법적 완성도를 평가하기 위해 정성적 분석을 실시하였다. 본 평가는 후처리 이전/이후 모델이 생성한 총 12편의 소넛을 대상으로 다음 네 가지 기준에 따라 비교하였다.

- 문법적 자연스러움 (Grammatical Coherence)
- 의미의 일관성 및 흐름 (Semantic Coherence)
- 시적 표현과 셰익스피어풍 스타일 유지 (Stylistic Fidelity)
- 반복/중복의 적절성 (Avoidance of Nonsensical Repetition)

항목	후처리 이전	후처리 이후
문법적 완성도	매우 낮음	향상됨
중복 및 반복 제어	심한 무작위 반복	다소 개선
의미 흐름	전반적으로 붕괴	일부 시에서 흐름 있음
셰익스피어풍 표현	드물게 유지	상대적으로 더 근접

후처리가 적용된 모델은 이전보다 정성적 평가에서 개선된 모습을 보였으나, 여전히 반복적 표현, 문법적 부자연스러움, 의미 단절 등의 문제가 존재한다. 이는 모델이 지역적 단어 생성에는 능하나, 전체 문맥을 유지하는 데는 한계가 있음을 시사한다. 향후 문맥 일관성을 강화하기 위해 prefix-tuning을 적용하거나, 전체 시 단위로 생성하도록 학습 데이터를 구성하는 방식을 도입할 수 있다. 또한, 시의 감정 흐름을

제어할 수 있는 컨트롤러 삽입을 통해 문학적 완성도를 개선할 수 있을 것으로 기대된다.

5. Discussion

5.1 감정 분석 (Sentiment Classification)

감정 분석 태스크에서는 리프레이징 및 joint-cascade 구조를 적용하였다. 리프레이징 기법은 클래스 간 데이터 불균형을 보완하고자 도입되었으며, 특히 ~ 그러나 joint-cascade 구조의 경우 SST와 같이 fine-grained한 감정 분류에서는 기대한 만큼의 성능 개선이 이루어지지 않았다. 이는 coarse 분류기의 오류가 후속 fine 분류기에 전이되며 누적되었을 가능성을 시사한다. 향후에는 dynamic routing 또는 confidence-aware 분기 구조를 적용함으로써 해당 한계를 보완할 수 있을 것으로 기대된다.

5.2 패러프레이즈 탐지 (Paraphrase Detection)

패러프레이즈 탐지에서는 학습 효율성과 성능 향상을 동시에 목표로 LoRA 및 Hard Negative Priority(HNP)를 적용하였다. LoRA는 학습 시간 단축과 GPU 메모리 절약에 효과적이었으며, HNP는 어려운 샘플을 중심으로 재학습을 시도하며 모델의 일반화 성능을 향상시키고자 하였다. 하지만 HNP 적용 이후 일부 성능 지표(F1-score, accuracy)에서 baseline보다 낮은 결과가 나타났으며, 이는 confidence threshold 설정의 민감성이나 HNP 데이터셋 내 노이즈의 영향을 받을 수 있음을 시사한다. 향후 threshold 최적화 및 noise filtering 전략이 필요하다.

5.3 소네트 생성 (Sonnet Generation)

소네트 생성 태스크에서는 rhyme 구조의 보존과 정량적 성능의 균형을 목표로 rhyme loss와 구조 보장형 후처리(postprocessing) 기법을 적용하였다. rhyme loss는 BERScore나 Perplexity에 크게 영향을 주지 않으면서도 rhyme accuracy를 개선시키는 데 기여하였다. 또한 생성 결과의 형식적 완성도를 보장하기 위한 14행 고정 후처리는 rhyme accuracy를 더욱 향상시키는 데 유의미한 역할을 하였다. 다만, rhyme loss의 적용은 모델 수렴 속도에 영향을 주며, 지나치게 강한 rhyme 제약은 의미 일관성 감소로 이어질 가능성도 존재한다. 향후에는 soft rhyme

constraint 또는 attention-based rhyme filtering 기법 도입이 고려될 수 있다.

6. Conclusion

본 연구에서는 GPT-2 기반의 사전학습 언어모델을 활용하여 감정 분석, 패러프레이즈 탐지, 소넷 생성이라는 세 가지 자연어처리 태스크를 통합적으로 수행하고, 각 태스크의 특성에 맞는 성능 향상 기법을 적용하여 효과를 검증하였다.

감정 분석 태스크에서는 데이터 리프레이징을 통해 클래스 불균형 문제를 완화하고, cascade 구조 기반의 joint 학습 모델을 설계하여 fine-grained 분류 성능을 향상시키고자 하였다. 패러프레이즈 탐지에서는 LoRA를 통해 학습 효율을 개선하고, Hard Negative Priority(HNP)를 통해 난이도 기반 재학습을 수행하였다. 소넷 생성에서는 rhyme loss와 구조적 후처리 로직을 통해 운율 보존과 시의 형식 완성도를 동시에 개선하였다.

실험 결과, 각 기법들은 일부 평가 지표에서 긍정적인 효과를 나타냈으며, 특히 rhyme loss와 후처리 기법은 생성 텍스트의 정성적 품질을 향상시키는 데 효과적이었다. 반면, cascade 구조의 오차 누적이나 HNP의 threshold 민감성과 같은 한계도 확인되었으며, 이러한 점은 향후 개선이 필요한 과제로 남는다.

향후에는 cascade 구조의 동적 라우팅, soft constraint 기반 rhyme 제어 등의 기법을 통해 성능의 안정성과 일반화 능력을 더욱 향상시킬 수 있을 것으로 기대된다.

7. References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017).

"[Attention is all you need.](#)" Advances in Neural Information Processing Systems, 30.

-> GPT-2의 근간이 되는 논문으로, 전체 프로젝트의 기본 모델 구조를 설명

[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019).

"[Language models are unsupervised multitask learners.](#)" OpenAI Technical Report.

-> GPT-2의 사전학습 특성과 언어모델 기반 multitask 수행 가능성을 논의

[3] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., ... & Rajamani, S. (2021).

"[LoRA: Low-Rank Adaptation of Large Language Models.](#)" arXiv preprint arXiv:2106.09685.

-> 학습 효율성을 위해 적용되는 LoRA 기법의 근거 제시

[4] Robinson, J., Chuang, C., Sra, S., & Jegelka, S. (2021).

"[Contrastive Learning with Hard Negative Samples](#)".

International Conference on Learning Representations (ICLR), 2021.

-> 하드 네거티브 샘플링(Hard Negative Sampling)을 비지도 contrastive learning 맥락에서 분석하고 제안

[5] Lau, J. H., Cohn, T., & Baldwin, T. (2018).

"[Deep-speare: A joint neural model of poetic language, meter and rhyme.](#)" ACL.

-> rhyme 평가 지표와 rhyme을 반영한 cross entropy 기반의 loss를 사용하며 구조(운율, 줄 수, 각운)까지 동시에 고려

[6] Jason Wei, Kai Zou (2019).

"[EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks](#)" *Conference on Empirical Methods in Natural Language*

Processing (EMNLP) Workshop,

-> 텍스트 변형을 통해 텍스트 분류 성능을 향상시켰다는 점을 제시
(rephrasing에 대한 영감을 제공)

[7] Sanh, Wolf & Ruder (2019)

“**A Hierarchical Multi-task Approach for Learning Embeddings from Semantic Tasks**”, AAAI 2019

→ 계층적 태스크 분리 기반 multi-task learning 구조를 제안하며, coarse-to-fine cascade 접근의 이론적 배경

[8] Quora Question Pairs Dataset,

<https://www.kaggle.com/competitions/quora-question-pairs/data>

[9] Stanford Sentiment Treebank (SST) Dataset,

<https://nlp.stanford.edu/sentiment/>

[10] CFIMDB Movie Review Dataset,

<https://ai.stanford.edu/~amaas/data/sentiment/>

[11] The Complete Works of William Shakespeare,

<https://www.gutenberg.org/ebooks/100>