NoisyTune 기반 미세조정을 이용한 한국어 텍스트 분류

이현민, 나승훈, 임준호, 김태형, 류휘정, 장두성
전북대학교, ETRI, KT

Ⅰ. 서론

오늘날의 언어모델은 크게 두 가지 과정을 거쳐 학습을 진행한다.
대규모 학습 데이터를 통해 언어의 전반적인 이해를 위한 사전학습(Pre-train)을 한 후, 세부 태스크에 맞춰 미세조정(Fine-Tuning)을 진행한다. 그러나 사전학습 데이터와 미세조정 데이터간의 간극으로 인해 언어모델이 오버피팅(overfitting)이 발생할 위험이 있다. NoisyTune는 사전학습된 언어모델에 미세한 노이즈(Noisy)를 통해 미세조정이 잘 일어나도록 한다. 본 논문은 위에 상기한 NoisyTune을 한국어 사전학습 언어모델인 KoBERT에 Klue 데이터 셋을 활용하여 효과를 검증한다.

본 논문의 기여점은 다음과 같다.

- 한국어 언어모델 KoBERT 에서의 NoisyTune 적용성 확인
- 한국어 Klue NLI, STS, TC 데이터 상의 정확도 성능 비교 확인

Ⅱ. 관련연구

NoisyTune

사전학습된 언어모델은 대규모의 레이블 되지 않은 데이터를 사용하여 학습하는데, 다운 스트림 태스크는 비교적 소규모의 레이블링된 데이터를 사용하여 학습하기 때문에 두 데이터 간의 간극으로 인해 오버피팅이 발생할 수 있다. 이 문제를 해결하기 위해 사전학습된 언어모델에 약간의 노이즈를 더하여 다운 스트림 태스크에 미세조정시에 정답을 더 잘 찾을 수 있도록 도와준다.

학습된 파라미터에 노이즈를 부여하는 수식은 다음과 같다.

- $W_i = W_i + U\left(-\frac{\lambda}{2}, \frac{\lambda}{2}\right) * std(W_i)$
- 파라미터 $[W_1, W_2, \cdots, W_N]$

NoisyTune 수식

사전학습된 언어모델에는 다양한 파라미터들이 각기 다른 특성과 값을 가지고 있다. 이러한 특성을 반영하여 동일한 noisy를 사용하기 보다 각 파라미터들의 분산을 반영하여 다른 noisy를 생성하도록 한다.

위 수식은 아래의 Pytorch 스타일의 코드만 추가하면 손쉽게 적용 가능하다.

```
for named, para in model.named_parameters():
    model.state_dict()[named][:] +=
    (torch.rand(para.size()) - 0.5)
    *noisy_lambda*torch.std(para)
```

NoisyTune 소스코드

NoisyTune은 미세조정 과정에서 추가적인 연산없이 성능향상을 이끌어 낼 수 있다는 것이 장점으로 꼽힌다. 하지만, 하이퍼 파라미터 값인 noisy_lambda 값을 수정하며 결과를 확인해야 한다는 점에서 추가적인 실험이 필요하다. 실험적으로 noisy_lambda 값은 0.1, 0.15 일때 성능이 가장 좋다고 알려져있다.

Ⅲ. 실험 세팅

모델 및 데이터

실험에는 KoBERT 모델을 사용하였다. KoBERT는 Transformer의 인코더로만 구성된 언어모델이고, 한국어 Wiki 데이터 문장 5M, 단어 54M를 활용하여 학습하였다.

데이터셋으로는 Klue-NLI, STS, TC 데이터를 사용하였다.
각 데이터의 train_size는 Klue-NLI: 24,998개, Klue-STS: 11,668개, Klue-TC: 45,678개 존재한다.
데이터 예시는 아래와 같다.

문장	레이블
전제: "흠잡을 것 하나도 없는 정말 좋은 숙소였어요" 가설: "맘에 드는곳 하나 없는 숙소였어요."	2
전제: "흠잡을 것 하나도 없는 정말 좋은 숙소였어요" 가설: "특히 인테리어가 가장 멋졌어요."	1
전제: "흠잡을 것 하나도 없는 정말 좋은 숙소였어요" 가설: "정말 완벽한 숙소였어요."	0

Klue-NLI 데이터셋 예시

문장	레이블
문장1: "일단 방은 사진이랑 똑같고 매우 깨끗합니다." 문장2: "우선, 방은 그림과 같고 매우 깨끗합니다."	1
문장1: "그래서 욕실바닥에 까는 용도로 사용 했습니다." 문장2: "나는 모든 관광지를 도보로 여행 했습니다."	0

Klue-STS 데이터셋 예시

문장	레이블
극적 역전승 도로공사 GS칼텍스 꺾고 2년 연속 챔프 우승	5
트럼프 미중 무역협상 1단계 합의 서명식 가질 것 종합	4

Klue-TC 데이터셋 예시

하이퍼 파라미터

미세조정에 사용한 하이퍼 파라미터 구성은 다음 표와 같다.

Epoch	Learning rate	Batch	Warmup	seed
5	7e-6, 5e-5	16	0.1	Random seed 6

Ⅳ. 실험 결과

KoBERT를 사용하여 Klue-NLI, STS, TC validataion 데이터를 사용하여 측정한 결과이다. 모든 실험은 위에 서술한 하이퍼 파리미터와 동일한 값으로 실험을 진행하였으며, seed값을 랜덤하게 변경하며 나온 6개의 결과값의 평균을 기록하였다.

model	λ	NLI	STS	TC
KoBERT	0	76.8	79.5	86.1
+ NoisyTune	0.1	77.5	80.0	86.2

실험결과 Accuracy

실험 분석

KoBERT에 NoisyTune 기법을 적용하여 Klue-NLI, STS, TC 데이터셋으로 Accuracy 값을 측정하였다. 기존 정확도 76.8, 79.5, 86.1에 비해 평균 0.4 정도 상승한 77.5, 80.0, 86.2를 얻었다. 이를 통해 사전학습 모델의 파라미터에 약간의 노이즈를 추가하는 것이 사전학습시 사용한 데이터와 미세조정에 사용한 데이터의 간극을 줄여 다운 스트림 태스크에서 성능이 향상되는 것을 확인할 수 있었다.