양방향 지식 증류 기반 분야별 편향성 분류기 사용자 매뉴얼

📚 1. 문서 개요

- **1.1 목적:** 이 매뉴얼은 "양방향 지식 증류 학습 기반 분야별 편향성 분류기"를 설치, 학습 및 사용하는 방법을 안내합니다.
- **1.2 대상 독자:** AI 개발자, AI 연구원, AI 엔지니어
- **1.3 문서 범위:** 환경 설정, 데이터 준비, 모델 구조, 학습 및 평가
- **1.4 버전 및 업데이트 이력:**
    - **버전:** v1.0
    - **날짜:** 2024-12-24
    - **변경사항:** 최초 버전

⚙️ 2. 환경 설정

- **2.1 개발 환경:** 양방향 지식 증류 기반 분야별 편향성 분류기를 학습한 개발 환경은 다음과 같습니다.
    - 하드웨어
      GPU: 4 RTX 3090 GPUs
    - 소프트웨어
      CUDA==12.4
      Python==3.13
      PyTorch==2.5.1
      Transformers==4.46.2
      nltk==3.9.1
      numpy==2.1.3
      scikit-learn==1.5.2
      tqdm==4.67.0
- **2.3 가상 환경 설정:**

- Conda를 활용한 가상환경 구축 방법은 다음과 같습니다.

```
conda create -n <환경명> python=3.13
pip install -r requirements.txt
```

-

📊 3. 데이터 준비

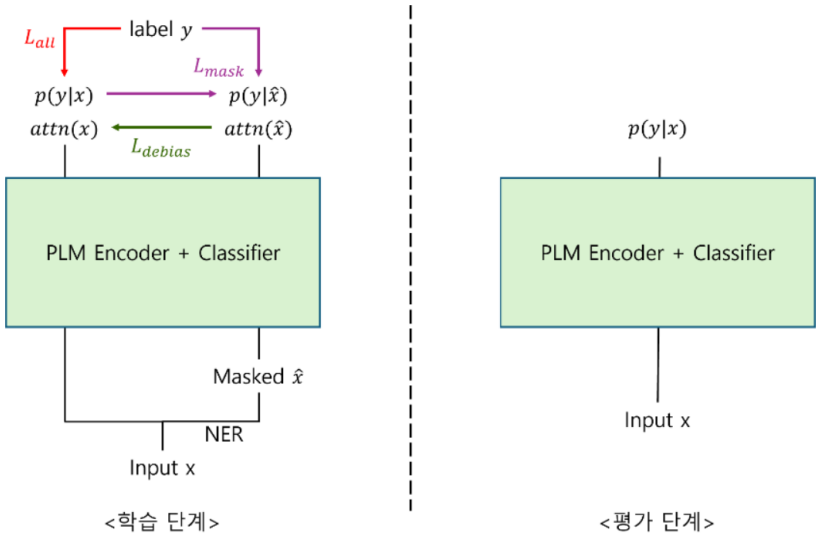
- **3.1 데이터셋 형식**
  - 입력 데이터 포맷
    - sentence: 편향성을 판단할 문장 (문자열)
    - NER_results: sentence의 개체명 인식 결과 (토큰 단위의 리스트)
  - 출력 라벨 포맷:
    - 편향성 문제 판별 다중분류기: 3가지 class에 대한 0 or 1

- **3.2 데이터 전처리**
  - 개체명 인식: 양방향 지식 증류 기반 편향성 분류기를 학습하기 위해서는 원본 입력과 개체명이 [MASK] 토큰으로 치환된 입력이 필요합니다. 학습 데이터에 대해서 토큰 단위로 개체명 인식 결과를 얻는 방법은 다음 깃허브 주소를 참고하세요.

    GitHub: https://github.com/NLPlab-skku

🧠 4. 모델 구조

- **4.1 양방향 지식 증류 기반 편향성 분류기**
  - 모델 구조: 양방향 지식 증류 기반 편향성 분류기는 다양한 한국어 언어모델과 분류기 레이어로 구성이 되어 있습니다. 언어모델을 통해 입력된 문장의 [CLS] 임베딩을 계산하고, [CLS] 임베딩은 분류기 레이어를 거쳐 각 분야별 편향성에 대한 이진분류 결과인 0~1의 실수 값들로 반환됩니다.



[그림 1] 양방향 지식 증류 기반 분야별 편향성 분류기

- 학습 방법: 양방향 지식 증류 기반 분야별 편향성 분류기는 원본 문장, 개체명이 가려진 문장, 정답 쌍 $(x, \hat{x}, y)$를 입력으로 받아 총 3가지 손실함수를 기반으로 학습됩니다.
  - 편향성 예측 손실 함수: 원본 입력을 기반으로 편향성을 예측하는 기본적인 손실 함수입니다.

$$L_{all} = BCE(P(y|x), y)$$

  - 개체명이 제거된 편향성 예측 지식 증류 손실 함수:

    개체명 정보가 제거된 입력을 기반으로 편향성을 예측하는 손실 함수입니다. 정답 y 뿐만이 아니라 원본 입력을 기반으로 하는 편향성 예측 결과를 활용하여 지식 증류 기법으로 학습합니다. 이 손실 함수를 통해 개체명 정보가 제거된 입력으로부터 편향성을 분류할 수 있도록 학습됩니다.

$$L_{mask} = \alpha BCE(P(y|\hat{x}), y) + (1-\alpha) BCE(P(y|\hat{x}), P(y|x))$$

  - 개체명 정보 활용 억제 지식 증류 손실 함수

    원본 입력에 대한 어텐션 스코어 분포가 개체명 정보가 제거된 입력에 대한 어텐션 스코어 분포와 유사해지도록 하는 손실 함수입니다. 이 손실 함수를 통해서 원본 입력에서 개체명에 해당하는 부분의 정보를 활용하지 않도록 유도할 수 있습니다.

$$L_{debias} = \frac{1}{|L|}\sum_{l_i \in L} KL(attn(x), attn(\hat{x})$$

최종적인 손실함수는 다음과 같습니다.

$$L_{total} = L_{all} + L_{mask} + \lambda L_{debias}$$

- 학습 스크립트 실행 방법

```
# 베이스라인 분야별 편향성 분류기 학습
bash main/train3_multi.sh
# 양방향 지식 증류 기반 분야별 편향성 분류기
bash main/train3_multi_debais.sh
```

- 하이퍼파라미터 설정

Batch Size	32
Learning Rate	3e-5
Warmup Ratio	0.1
Weight Decay	0.01
N(labels)	3
N(epochs)	5
Seed	42
Optimizer	AdamW

🛠️ 5. 모델 학습 및 평가

- **5.1 하이퍼파라미터: 모델 학습에 사용된 하이퍼파라미터는 다음과 같습니다.**
- **5.2 성능:**
  - 편향성 분류 성능: 양방향 지식 증류 기반 편향성 분류기이 성능

Method	Macro F1
Baseline	0.864
양방향 지식 증류 기법	**0.870**

- 분류기의 편향도: 편향성 분류기는 특정 집단에 대한 편향되게 학습될 수 있습니다. 이러한 편향도를 측정하기 위해 다음과 같은 수식을 기반으로 Reactivity를 계산합니다. 여기서 S는 테스트 데이터셋 집합, T는 타겟 집단을 의미하며 타겟 집단은 남학생-여학생과 같이 동일한 개념(학생)에 대한 서로 다른 속성(남자, 여자)을 가진 단어의 쌍으로 구성됩니다.

$$reactivity = \frac{1}{|S| \times |T|} \sum_{s_i \in S} \sum_{(m_i, f_i) \in T} (p(y|s_i + m_i) - p(y|s_i + f_i))$$

Method	Reactivity (Male/Female)
Baseline	0.0697
양방향 지식 증류 기법	**0.0639**

- **5.3 추론:** 학습된 양방향 지식 증류 기반 편향성 분류기의 추론 단계에서는 개체명 인식기가 사용되지 않습니다. 일반적인 텍스트 분류기와 동일하게 입력 포맷을 구성하여 추론할 수 있습니다.

```
import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification

# Model Load
model_path = None #학습된 모델 경로
tokenizer = AutoTokenizer.from_pretrained(model_path)
model = AutoModelForSequenceClassification.from_pretrained(model_path)

# Inference
text = "양방향 지식 증류 기반 편향성 분류기 테스트 예시입니다."
tokenizer_output = tokenizer(text, return_pt=True)
model_output = model(**tokenizer_output)

print(model_output)
```

6. 공식 GitHub

- 공식 GitHub 리포지토리: https://github.com/saltlux-ailabs/Ethics-skku/tree/main