# KorSciDeBERTa 환경 설치 & 파인튜닝

### [참고] Deberta-native (github)

https://github.com/microsoft/DeBERTa

### KorSciDeBERTa-native 설치

- 추후 모델 제출시 결과 재현을 위해 아래 환경 파일 생성 후 동봉 요망

1. conda env export > deberta.yaml
2. pip freeze > requirements.txt

```
#아래 명령을 순서대로 실행하면서 에러 확인
git clone https://huggingface.co/kisti/korscideberta; cd korscideberta; unzip
korscideberta.zip -d korscideberta; cd korscideberta

conda create -n deberta python=3.8 --quiet --yes
#conda init bash; source ~/.bashrc
#파이썬 3.8~3.9(3.10 미지원). torch 1.10(1.13이상 미지원)
conda activate deberta; pip3 install -r requirements.txt; pip install --upgrade nltk;
pip uninstall -y torch torchtext torch-tensorrt; pip install --upgrade pip; pip install
torch==1.10.1+cu111 torchvision==0.11.2+cu111 torchaudio==0.10.1 -f
https://download.pytorch.org/whl/cu111/torch_stable.html --default-timeout=100; pip
install setuptools_scm six mlflow; pip install "numpy<1.24.0"; pip install .

#pip설치 안될 시 kakao 서버 이용
#pip install six mlflow -i http://ftp.daumkakao.com/pypi/simple --trusted-host
ftp.daumkakao.com; pip install "numpy<1.24.0" -i http://ftp.daumkakao.com/pypi/simple --
trusted-host ftp.daumkakao.com; pip install -r requirements.txt -i
http://ftp.daumkakao.com/pypi/simple --trusted-host ftp.daumkakao.com; pip install --
upgrade nltk i http://ftp.daumkakao.com/pypi/simple --trusted-host ftp.daumkakao.com;
pip install .
```

### mecab 설치

-

- cd mecab
- bash <(curl -s https://raw.githubusercontent.com/konlpy/konlpy/master/scripts/mecab.sh); cd mecab-0.996-ko-0.9.2;
- chmod 775 ./configure; ./configure; make; chmod 775 tests/*.sh; make check; make install

    (권한 에러시 sudo 사용 버전) sudo chmod 775 ./configure; ./configure; make; sudo chmod 775 tests/*.sh; sudo make check; sudo make install

    -

    make 에러 발생 사례 및 해결

        - (make 에러 및 해결법)

```
에러 1:
libtool: Version mismatch error.  This is libtool 2.4.2 Debian-2.4.2-1ubuntu1, but the
libtool: definition of this LT_INIT comes from libtool 2.4.6.
libtool: You should recreate aclocal.m4 with macros from libtool 2.4.2 Debian-2.4.2-
1ubuntu1
libtool: and run autoconf again.

해결법:
autoreconf --force --install; ./configure; make

에러 2:
configure.in:23: error: required file './compile' not found
configure.in:23:   'automake --add-missing' can install 'compile'
configure.in:6: error: required file './missing' not found
configure.in:6:   'automake --add-missing' can install 'missing'
```

- (make 에러 2 발생시 아래를 실행 후 재시도)
- 해결법
- apt-get install automake perl; apt-get update; apt-get upgrade; apt install build-essential automake dh-autoreconf libusb-1.0-0-dev cmake g++; apt-get install libtool; automake --add-missing; autoreconf; autogen.sh; make clean
- (sudo 사용 버전) sudo apt-get install automake perl; sudo apt-get update; sudo apt-get upgrade; sudo apt install build-essential automake dh-autoreconf libusb-1.0-0-dev cmake g++; sudo apt-get install libtool; automake --add-missing; autoreconf; autogen.sh; make clean
- 대처 이후에도 에러 사례

```
make[2]: *** [Makefile:559: install-libLTLIBRARIES] Error 1
make[2]: Leaving directory '/home/work/DeBERTa/mecab/mecab-0.996-ko-0.9.2/src'
make[1]: *** [Makefile:761: install-am] Error 2
make[1]: Leaving directory '/home/work/DeBERTa/mecab/mecab-0.996-ko-0.9.2/src'
make: *** [Makefile:515: install-recursive] Error 1
```

    - #Cmake 없는 경우 설치 방법
        - https://mong9data.tistory.com/124

- ###2. mecab-ko-dic 설치
- cd ../mecab-ko-dic-2.1.1-20180720; chmod 775 ./autogen.sh; ./autogen.sh; ./configure; make
- (sudo 사용 버전) cd ../mecab-ko-dic-2.1.1-20180720; sudo chmod 775 ./autogen.sh; ./autogen.sh; ./configure; make
    - 갱신할 내용이 없는 경우에는 "make: Nothing to be done for 'all'." 출력됨
- ###3. 사용자 사전파일을 user-dic 폴더로 복사 & 설치
- cp ../pa* ./user-dic/; chmod 775 ./tools/add-userdic.sh; ./tools/add-userdic.sh; make install
- (sudo 사용 버전) cp ../pa* ./user-dic/; sudo chmod 775 ./tools/add-userdic.sh; ./tools/add-userdic.sh; sudo make install
    - 수 분 소요됨

- ###4. 설치 확인법
- mecab -d /usr/local/lib/mecab/dic/mecab-ko-dic
- 곧바로 콘솔에 '원천기술' 타이핑하여 입력시 원천 / 기술로 나누어지지 않고 원천기술로 출력되면 정상 설치(Ctrl+C로 나가기)
- 구동부 입력시 구동/부 가 아닌 구동부
- cd ../..
- ###5. 설치 중/후 문제 발생시 아래 명령어 실행 후 상기 설치 재확인
- pip install mecab-python3; apt-get install mecab mecab-ipadic-utf8 libmecab-dev swig; bash <(curl -s https://raw.githubusercontent.com/konlpy/konlpy/master/scripts/mecab.sh)
- (sudo 사용 버전) pip install mecab-python3; sudo apt-get install mecab mecab-ipadic-utf8 libmecab-dev swig; bash <(curl -s https://raw.githubusercontent.com/konlpy/konlpy/master/scripts/mecab.sh)

## 학습

### 주제분류 학습

conda activate deberta
cd korscideberta/experiments/glue; chmod 777 *.sh;

(학습 실행) ./mnli.sh
: mnli.sh 내에서 **tag**=1575000ntis1tier 에 학습된 모델 등을 출력할 폴더명을 지정

```
11 #set output folder name
12 tag=1575000ntis3tier
```

(모델 경로) init="checkpoint0324/pytorch.model.bin"
(최신 체크포인트) init="korscideberta/pytorch_model.bin"
(OOM시) train_batch_size를 64 → 32로 변경하여 해결

(입력) glue_tasks/MNLI/train.tsv, test_matched.tsv, dev_matched.tsv
: glue_tasks 하위폴더에 MNLI-ntis3tier(소분류), MNLI-ntis2tier, MNLI-ntis1tier(대분류)와 같이 학습 데이터가 준비되어 있고,
**실제 학습은 'MNLI'폴더 데이터가 이용**되므로 학습할 데이터 폴더를 'MNLI'로 수동으로 바꾸어 주어야 함.
10-fold를 구현하려면 10개의 학습 데이터를 준비해 놓고, 기준 'MNLI'폴더 대신 10-fold 학습데이터 폴더를 조회하도록 변경해야 함.
(입력 데이터 설명)

```
premise, hypothesis = 원본 mnli태스크에서 문장1, 문장2
현 태스크에서는
premise = 제목+저널명+서론
hypothesis = 빈칸

RCMN = 정답 레이블,
top1-3 = 예측 레이블
label = 분류 학습할때 쓰이는 라벨명(대/중/소 선택)

label2tier = 사용하지 않음
acc = 3개중에 2개 맞았으면 2
```

(출력-모델 파일) out/157500ntis1tier/pytorch.model-001673.bin
(학습 스크린샷)

```
06/16/2023 16:25:16|INFO|MNLI|00| Loading labels: /media/hdd1/kkm/DeBERTa-fine/experiments/glue/NTIS/tier1/6/train.tsv
06/16/2023 16:25:27|INFO|MNLI|00| Labels: [' ', 'EA', 'EB', 'EC', 'ED', 'EE', 'EF', 'EG', 'EH', 'EI', 'HA', 'HB', 'HC', 'HD', 'HE', 'LA', 'LB', 'LC', 'NA', 'NB', 'NC
', 'ND', 'OA', 'OB', 'OC', 'SA', 'SB', 'SC', 'SD', 'SE', 'SF', 'SG', 'SH', 'SI', '촉매']
06/16/2023 16:25:29|INFO|MNLI|00| Total corpus examples: 1197
06/16/2023 16:25:29|INFO|MNLI|00| Total corpus examples: 20
06/16/2023 16:25:29|INFO|MNLI|00|   Evaluation batch size = 256
06/16/2023 16:25:32|INFO|MNLI|00| Total corpus examples: 1196
06/16/2023 16:25:32|INFO|MNLI|00| Total corpus examples: 20
06/16/2023 16:25:32|INFO|MNLI|00|   Prediction batch size = 32
```

```
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]), dtype=torch.int32)), ('labels', tensor(14, dtype=torch.int32))])
/home/kgm86/anaconda3/envs/deberta/lib/python3.9/site-packages/DeBERTa/optims/xadam.py:42: UserWarning: This overload of add_ is deprecated:
        add_(Number alpha, Tensor other)
Consider using one of the following signatures instead:
        add_(Tensor other, *, Number alpha) (Triggered internally at  ../torch/csrc/utils/python_arg_parser.cpp:1050.)
  next_m.mul_(beta1).add_(beta1_, grad)
06/16/2023 16:46:11|INFO|MNLI|00| None[0.7%][-3.13h] Steps=100, loss=3.4740912413597105, examples=3200, loss_scale=16384.0, 84.3s
```

**주제분류 추론, 평가**

**추론**

(추론 실행) mnli-pred.sh

: 추론 시행 후, 학습시와 동일하게 tag= 에서 지정한 출력폴더에 레이블별 추론 확률을 저장함.

(입력) 테스트 데이터 glue_tasks/MNLI/test_matched.tsv

(확률값 출력) out/1575000ntis1tier-test/test_logits_matched_1575000ntis1tier-test.txt

: out/$tag/test_logits_matched_$tag.txt

test_logits_matched_1575000NTIS_TIER1_KSC_FOLD9.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
-5.4785156250000000e-01  -1.2041015625000000e+00  -1.4433593750000000e+00
-3.2910156250000000e+00  1.6279296875000000e+00  -6.1865234375000000e-01
-2.0644531250000000e+00  6.8085937500000000e+00  -5.3027343750000000e-01
```

**[첨부]**

**평가 준비**

(평가 준비) eval/3testlogitstsvTestonlyTotop3.ipynb

: 예측 확률값을 예측 레이블로 바꿈

(입력) tier = 1(대분류)

테스트 데이터: filename = 'MNLI-ntis1/test_matched.tsv'

학습했던 데이터: trainlabelfile = 'MNLI-ntis1/train.tsv'

확률값 파일: probfile = 'MNLI-ntis1/test_logits_matched_1575000ntis1tier-test.txt'

(출력)

outfile = 'MNLI-ntis1/20230607_debertaTier1test.tsv'

: tsv 파일로, 다음 컬럼에 정답 레이블과 예측 레이블 코드 리스트가 출력됨. [5,6,7], [12,13,14]

(출력 예)

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | NTNTS_C | YEAR | PER_NM_K | NL_NM_K | APER_TEX | CMN_CD1 | CMN_CD2 | CMN_CD3 | sentence1 | sentence2 | 1tier | label1 | top1 | top2 | top3 |
| 2 | JAKO2013 | 2013 | 철도시스템 | E2M - 전기 | 서론 나노 | EC0204 | EI0602 | EI0601 | 활성탄 함유량에 따른 광촉매(TiO<sub>2 | | | | EA1202 | EF0601 | EB0304 |
| 3 | JAKO2013 | 2013 | 방사선의로 | E2M - 전기 | 서론 지난 | EG0704 | LC0412 | LC0405 | 인자화된 최대 공산선형회귀 적응기법을 | | | | EG0704 | EG0705 | EG0407 |

**Klue/Glue 벤치마크 학습**

cd experiments/glue;

chmod 777 *.sh; export CUDA_VISIBLE_DEVICES=0; ./stsb-glue.sh
chmod 777 *.sh; ./mnli.sh
chmod 777 *.sh; export CUDA_VISIBLE_DEVICES=0,1; ./ner-dp.sh
chmod 777 *.sh; export CUDA_VISIBLE_DEVICES=0,1; ./record.sh
chmod 777 *.sh; export CUDA_VISIBLE_DEVICES=0,1; ./cola.sh
chmod 777 *.sh; export CUDA_VISIBLE_DEVICES=0,1; ./cola.sh