maum Academy
AI Professional Education

Level 2. How to customize AI engines

http://mindslab.ai

1. XDC 개요

1. 데이터 정제

1. XDC Train / Inference

1. XDC 실습

maum Academy
AI Professional Education

XDC

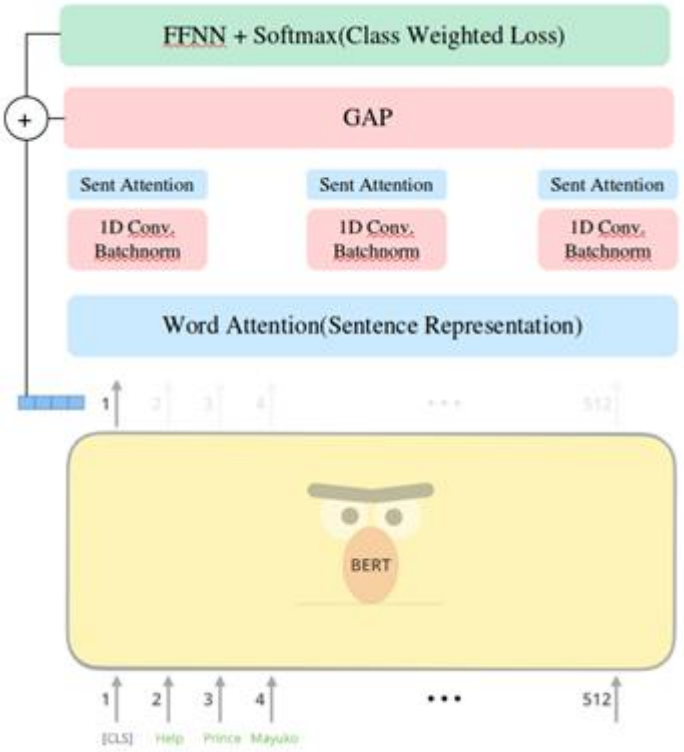
eXplainable Document Classifier

XDC 란

- Google에서 발표한 BERT 아키텍처 위에 CNN attention layer를 더해 텍스트 분류를 수행한다. 텍스트 분류의 근거가 되는 문장과 단어의 위치를 확인할 수 있는 '설명 가능한' 문서 분류 엔진이다.

- BERT는 Transformer Encoder 구조를 가진 사전학습 + fine-tuning 네트워크이다

- 대량의 텍스트 데이터를 활용해 비지도(unsupervised) 사전학습을 수행하고, 수행 task에 따른 fine-tuning 과정을 거친다.



maum Academy
AI Professional Education

XDC

XDC source tree

Source tree

- core/: xdc_modules
  - modeling: BERT 네트워크
  - optimization: Adam Optimizer
  - tokenization: BERT tokenizer
  - utils: 텍스트 전처리
  - xdc_core: fine-tuning 수행
  - data/: 데이터 폴더
    - raw: 원본 파일 보관용. 학습시엔 불필요

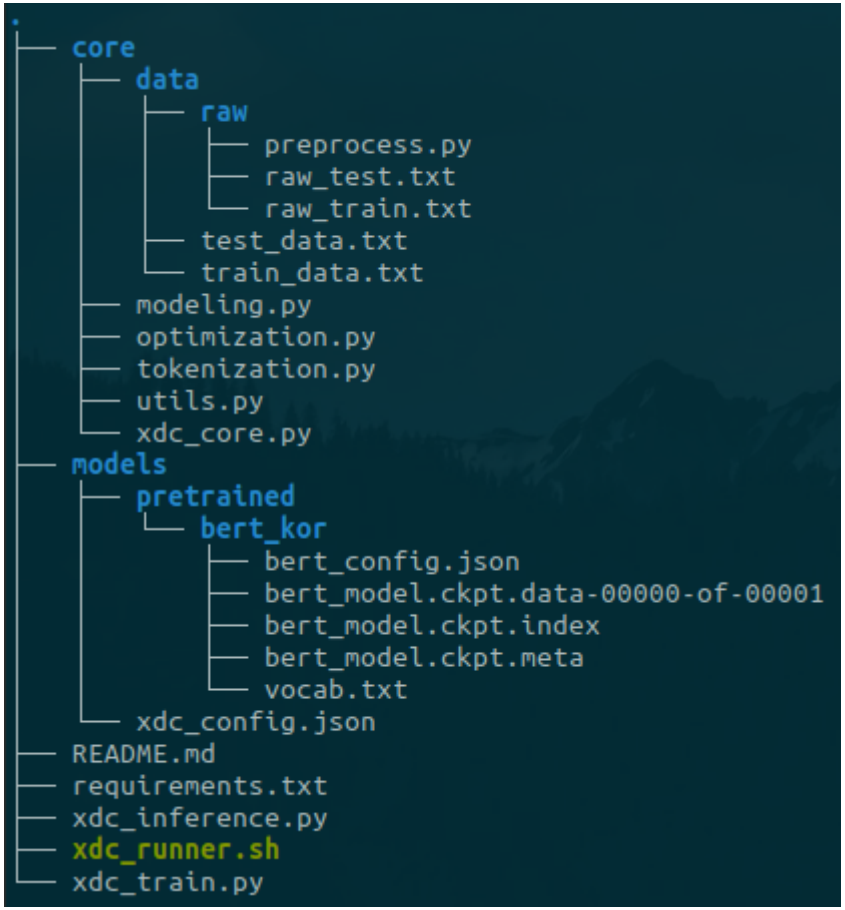
- models/: 학습 후 모델이 생성되는 위치
  - pretrained/: 사전 학습 모델 위치

- xdc_inference.py: 학습 후 결과 확인
- xdc_train.py: 학습(or 테스트) 코드

핵심 File

- xdc_runner.sh: 학습 실행 스크립트

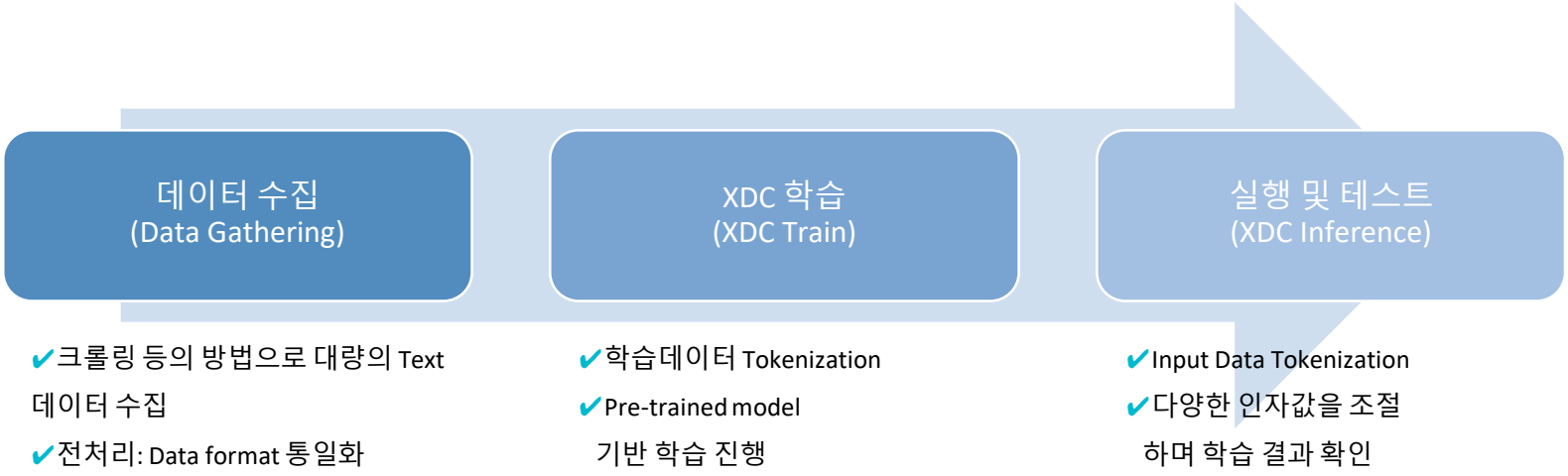
- models/xdc_config.json: 모델 경로 등 인자 설정



* XDC가 동작하기 위한 기본 서버 환경 구조

XDC 학습 및 실행 Process

- 기본적으로 아래와 같이 3단계로 진행하며, 성능 안정화가 될 때까지 반복 작업
- 사전학습은 다량의 데이터와 오랜 시간이 필요하기 때문에 미리 학습된 모델을 활용



1. XDC 개요

1. 데이터 정제

1. XDC Train / Inference

1. XDC 실습

maum Academy
AI Professional Education

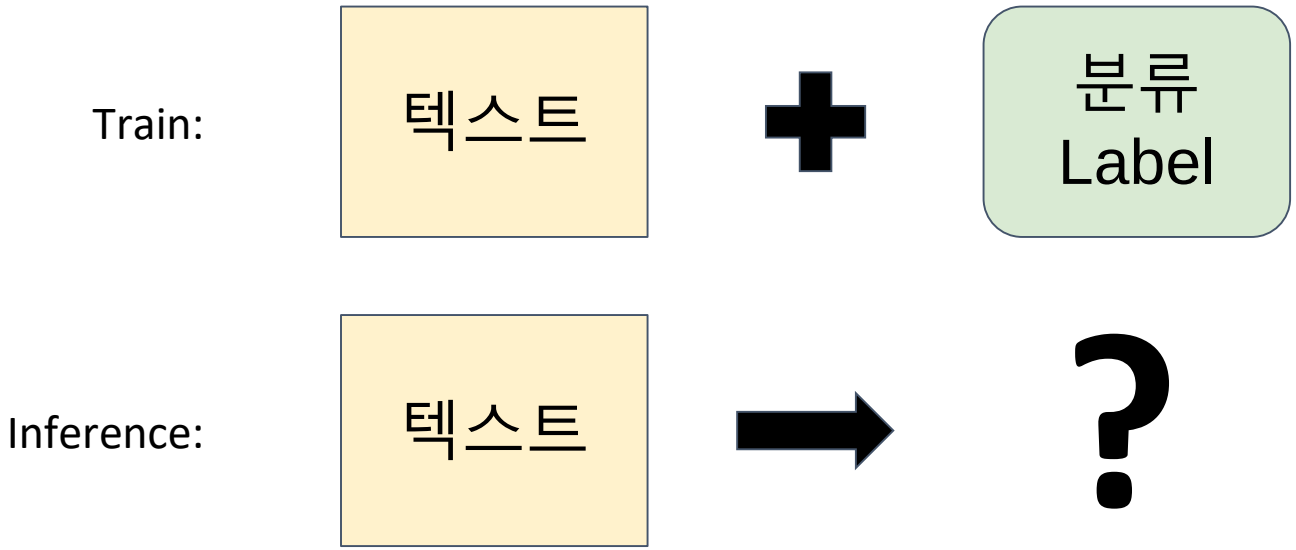
XDC 학습 데이터

XDC Dataset

데이터 유형

- XDC 엔진은 1) 텍스트와 2) 해당 텍스트에 대한 분류를 제공해 이를 학습하고, 이후 텍스트를 입력했을 때 해당 텍스트의 분류를 추론한다.

- 예를 들어 아래와 같은 데이터가 XDC 학습에 적합하다.
  - 1) 영화 리뷰 텍스트 + 2) 긍/부정 평가
  - 1) 뉴스 기사 텍스트 + 2) 뉴스 주제
  - 1) 메일 본문 텍스트 + 2) 스팸 여부
  - 1) 콜센터 대화 내용 + 2) 불만/비불만 분류



maum Academy
AI Professional Education

XDC 학습 데이터

XDC Dataset

데이터 범위

- 지원 언어: 사전 학습 모델을 달리 하여, 한국어 / 영어 / 다국어(독일어, 프랑스어, 중국어 등 100개 언어) 지원

- 데이터 길이: 일반적 길이의 25문장(4~5문단) 정도까지 입력 가능하다. BERT 사전 학습이 토큰(한글의 경우 형태소) 기준 512개까지 됐기 때문. 아래 예시 기사의 토큰 길이가 510이다.

범용 인공지능 플랫폼을 만든 마인즈랩, 유통 모든 과정에 AI솔루션을 도입한 쿠팡, AI스피커 '누구'를 만든 SK텔레콤 등 14개 업체가 '2019 대한민국 인공지능대상' 부문별 대상을 받았다. 또 마인드에이아이 등 5개 업체가 스타트업 부문상을, 육군교육사 전투발전부와 진주시청 등 4개 기관과 기업이 특별상을 수상했다. '2019 대한민국 인공지능대상' 시상식이 5일 웨스틴조선 오키드룸에서 열렸다. 인공지능대상은 우리나라 인공지능(AI)산업 생태계의 발전을 촉진하고 이를 통해 글로벌 경쟁력을 확보할 수 있도록 돕기 위해 IT조선과 마이크로소프트웨어가 만든 상이다.

무엇보다 우리나라 AI 기술과 산업 경쟁력이 미국, 중국은 물론이고 유럽연합(EU)과 일본보다 뒤처졌다는 위기감과 절박감이 이 상을 제정하게 끔 만들었다. AI 부문 대상을 만든 것은 한국은 물론이고 아시아 처음이다. AI선진국과의 격차를 하루빨리 좁혀 우리나라가 미래 디지털 혁명시대도 선도하자는 의지의 반영이다. IT조선은 올해 첫 시상을 시작으로 매년 확대 발전시켜 숨겨진 AI 전문업체까지 지속적으로 발굴해 산업 발전에 일조할 계획이다. 조선미디어그룹이 후원하며, 과학기술정보통신부와 정보통신산업진흥원, 한국정보화진흥원 등 정부 기관도 여기에 힘을 보탰다.

영예의 첫 대상을 모두 23개 기업과 기관이 수상했다. 14개 기업이 부문별 대상을, 5개 기업이 스타트업 대상을 받았으며, 4개 기업과 기관이 특별상을 수상했다. IT조선은 우선 국내외 인공지능(이하 AI) 시장과 기술 동향을 조사해 ▲제조 ▲서비스 ▲유통 ▲금융 ▲의료바이오 ▲컴퓨팅 ▲교육 ▲공공지자체 등 부문별 기업을 선별해 수상자를 선정했다.

2019 대한민국 인공지능대상 유통부문은 축산물 직거래 유통 플랫폼 '미트박스'를 운영하는 글로벌네트웍스가 수상했다. 법률 부문 수상자 인텔리콘연구소는 사건 판례, 법령을 인공지능 분석해 설명을 덧붙이는 '지능형 검색기'를 선보였다. 문화콘텐츠 부문 수상자는 동양온라인이다. 세계 최대 바둑포털 '타이젬'으로 유명한 회사다. 마인즈랩은 컴퓨팅 부문 상을 받았다. 음성과 언어, 시각과 교육 등 다방면에 응용할 수 있는 AI 플랫폼 '마음아이'를 내놨다.

...

(11/5 IT조선 기사 중)

XDC 데이터 준비

XDC Data preparation

데이터 구축

- 지도 학습(Supervised Learning)을 위해 텍스트와 함께 분류 정보가 필요하다.
    1. 분류 정보가 존재하는 경우(뉴스 기사 + 주제, 영화 리뷰 + 평점 등)
       → 기존에 있는 데이터를 크롤링 등으로 수집해서 만들 수 있다.
    2. 분류 정보가 존재하지 않는 경우(발화 내용 + 감정 등)
       → 인력을 동원해 텍스트에 분류를 더해서 데이터를 구축한다.

- 간단한 실험 및 연습을 위해서 공개된 데이터를 활용할 수 있다. (네이버 영화 리뷰, IMDb 등).

ex) 네이버 영화 리뷰(nsmc) 데이터 다운로드 예시(linux)

```
$ wget https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt
$ wget https://raw.githubusercontent.com/e9t/nsmc/master/ratings_test.txt
```

ex) ratings_train.txt (nsmc 학습 데이터. 0: 부정, 1: 긍정)

```
id  document    label
9976970 아 더빙.. 진짜 짜증나네요 목소리    0
3819312 흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나    1
10265843    너무재밋었다그래서보는것을추천한다  0
9045019 교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정   0
```

XDC 데이터 준비

XDC Data preparation

데이터 형식: [Text] + '\t' + [Label] + '\n' ...

- [Text]: 분류할 텍스트 데이터.
- '\t': 텍스트/레이블 구분자로 tab을 사용한다.
- [Label]: 분류명을 자연어 그대로 사용 가능하다. 학습 과정 중 내부적으로 정수로 변환하여 처리한다.
- '\n': 각 데이터 행을 구분한다.

데이터 정제

- header나 index는 문서에 포함되지 않도록 한다
- 구분자는 '\t'로 하여 저장한다

ex) preprocess_nsmc.py

```python
import pandas as pd

raw_paths = ['ratings_train.txt', 'ratings_test.txt']
result_paths = ['train_data.txt', 'test_data.txt']

for raw_path, result_path in zip(raw_paths, result_paths):
        raw = pd.read_csv(raw_path, sep='\t')
            raw.to_csv(result_path, columns=['document', 'label'], sep='\t', header=False, index=False)
```

ex) train_data.txt (nsmc 학습 데이터 정제 후)

```
아 더빙.. 진짜 짜증나네요 목소리    0
흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나    1
너무재밓었다그래서보는것을추천한다  0
교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정    0
```

1. XDC 개요

1. 데이터 정제

1. XDC Train / Inference

1. XDC 실습

XDC Train

XDC Train Arguments

xdc_train.py 명령 옵션

```
parser.add_argument('--model_name', type=str, default='tmp')
parser.add_argument('--num_train_epochs', type=float, default=5)
parser.add_argument('--train', action='store_true', default=False)
parser.add_argument('--en', action='store_true', default=False)
```

- model_name: 학습 모델명. (ex: nsmc_sentiment)

- num_train_epochs: epoch 횟수(학습 데이터 반복 학습 횟수)

- train: train / test 조정 플래그. 설정하면 train, 생략하면 test로 실행

- en: en / ko 조정 플래그. 설정하면 영어, 생략하면 한글로 실행

models/xdc_config.json 환경 설정

```
"model_dir": "/models/",
"vocab_file": "/models/pretrained/bert_kor/vocab.txt",
"init_checkpoint": "/models/pretrained/bert_kor/bert_model.ckpt",

"train_data": "core/data/train_data.txt",
"dev_data": null,
"test_data": "core/data/test_data.txt",
```

- model_dir: 학습 모델 저장 위치 (맨 앞 / 포함)
- vocab_file: 사전 학습 결과 생성되는 vocabulary file의 경로
- init_checkpoint: 사전 학습 모델의 체크포인트 경로
- train_data: 학습 데이터 경로 (맨 앞 / 생략)
- test_data: 테스트 데이터 경로

models/xdc_config.json 환경 설정

```
"num_top_label": 3,
"num_top_word": 3,
"num_top_sent": 3,

...

"xdc_checkpoint": "/models/{MODEL_NAME}"
```

- num_top_label: 결과로 확인할 추측 레이블의 수
- num_top_word: 추측 레이블 당 추출할 근거 단어의 수
- num_top_sent: 추측 레이블 당 추출할 근거 문장의 수
- xdc_checkpoint: 학습된 모델의 저장 위치

XDC Train

Run XDC Train

학습 command 예시

```
python ./train.py \
     --model_name='nsmc_sentiment'
     --num_train_epochs=5.0
     --train
```

xdc_runner.sh 스크립트에 실행을 위한 명령어가 입력돼 있으며, 필요한 인자를 조정한 후, (실행 권한 부여 후) 실행하면 된다. 스크립트 내의 export CUDA_VISIBLE_DEVICES 는 사용할 GPU의 id를 지정해주는 것이다.

테스트 command 예시

```
python ./train.py \
     --model_name='nsmc_sentiment'
```

```
Acc: 0.7940

Classification Report
              precision    recall  f1-score   support

           0       0.78      0.79      0.78       238
           1       0.80      0.80      0.80       262

    accuracy                           0.79       500
   macro avg       0.79      0.79      0.79       500
weighted avg       0.79      0.79      0.79       500
```

models/{MODEL_NAME}/test_results_N.txt 에 테스트 결과 저장 됨.
위 결과는 nsmc 전체 데이터의 1%만 추출하여 학습(1000건) 및 테스트(500건)한 결과

XDC Inference

XDC Inference Arguments

xdc_inference.py 명령 옵션

```
parser.add_argument('--lang', type=str, default='ko')
args = parser.parse_args()

config = json.load(open(os.curdir+'/models/xdc_config.json', 'r'))
model = BertXDC(config, args.lang)

context = "전주맛집 중 빠질 수 없는 전주 먹거리는 바로 막걸리다. "\
          "전주에서 막걸리 하면 상다리가 휠만큼 많은 안주들을 맛볼 수 있으며 푸짐한 인심으로도 유명하다. "\
          "그중 맛은 물론 합리적인 가격과 분위기, 볼거리가 많은 \"한옥전주막걸리\"는 "\
          "넉넉한 인심과 친절한 서비스를 자랑한다. "\
          "가족과의 외식, 회식장소로도 많은 관심을 받고 있는 \"한옥전주막걸리\"는 "\
          "전체 인원 200명 수용 가능한 내부 크기와 야외 테라스 및 최신 트렌드에 걸맞는 젊은층을 "\
          "겨냥한 인테리어를 자랑하고, 10가지 이상의 기본 안주는 물론, 메인 안주에도 노력을 쏟아 "\
          "방문객들의 재방문이 줄을 잇고 있다. 매일 시간마다 다른 구성으로 즐길 수 있는 색다른 라이브 공연으로 "\
          "눈과 귀를 만족시켜주는 장점을 갖춰 다른 막걸리집과 차별함을 두고 있으며, 다양한 볼거리가 많은 "\
          "전주한옥마을에서 5분 정도의 가까운 거리에 위치해있어 전주를 찾는 관광객들에게도 많은 관심을 받고 있다. "\
          "젊은층에게도 인기 있는 \"한옥전주막걸리\"는 언제나 건강한 재료를 사용하고 남녀노소 즐길 수 있는 "\
          "막걸리를 널리 알리기 위해 전 직원이 노력하고 있다고 관계자는 밝혔다. 전라북도 블로거들이 적극 추천하는 "\
          "인증 먹거리 \"한옥전주막걸리\"는 전주 우아동에 위치하고 있으며 오후 4시부터 오전 2시까지 영업하고 있다. "\
          "공연 및 단체 예약문의는 063-241-4466으로 가능하다."
```

- lang: 학습 언어 (영어: en, 한글: ko)

- context: 분류할 문장. 텍스트 에디터로 코드 내 수정

maum Academy
AI Professional Education

XDC Inference

Run XDC Inference

추론 command 예시

```
python ./xdc_inference --lang=’ko’
```


```
[[[('1', 0.7257912755012512), ('0', 0.2742087244987488)], [([425, 428], '거리에', 1.0), ([515, 517], '있는', 3.1721248e-34), ([615, 619], '4시부터', 4.9287175e
-35)], [[[150, 313], '가족과의 외식, 회식장소로도 많은 관심을 받고 있는 "한옥전주막걸리"는 전체 인원 200명 수용 가능한 내부 크기와 야외 테라스 및 최신 트렌드에
 걸맞는 젊은층을 겨냥한 인테리어를 자랑하고, 10가지 이상의 기본 안주는 물론, 메인 안주에도 노력을 쏟아 방문객들의 재방문이 줄을 잇고 있다.', 0.0335916653275489
8], [[463, 557], '젊은층에게도 인기 있는 "한옥전주막걸리"는 언제나 건강한 재료를 사용하고 남녀노소 즐길 수 있는 막걸리를 널리 알리기 위해 전 직원이 노력하고 있
다고 관계자는 밝혔다.', 0.03350228816270828], [[557, 636], '전라북도 블로거들이 적극 추천하는 인증 먹거리 "한옥전주막걸리"는 전주 우아동에 위치하고 있으며 오후
 4시부터 오전 2시까지 영업하고 있다.', 0.033522073179483414]]]]
```

1. XDC 개요

1. 데이터 정제

1. XDC Train / Inference

1. XDC 실습

실습 주제

- 개발환경 준비
- 데이터 셋 준비
- XDC 엔진 학습
- XDC Inference 확인

1. 개발 환경 접속

1) 서버에 접속

- Host : 192.168.0.1
- User : xdc-exercise
- Password : xxxxxx

2) XDC가 설치된 위치로 이동 (이하, '**기본 위치**'라 함)

```
cd /home/xdc-exercise/workspace/xdc
```

2. 학습 데이터 준비

1) 학습데이터를 준비하여 아래의 위치에 옮겨놓는다.

```
/home/xdc-exercise/xdc/core/data/train_data.txt
/home/xdc-exercise/xdc/core/data/test_data.txt
```

데이터 위치 경로 데이터 파일

2) 필요한 경우 데이터 전처리를 수행한다 ('**데이터 정제**' 섹션 참고)

3. XDC Train

학습데이터 경로 등 Arguments를 정확히 설정하여 학습 명령어 실행

*실행 위치 : 기본 위치

Arguments 설정 예시(models/xdc_config.json)

```
"model_dir": "/models/",
"vocab_file": "/models/pretrained/bert_kor_mecab/vocab.txt",
"init_checkpoint":
"/models/pretrained/bert_kor_mecab/bert_model.ckpt",

"train_data": "core/data/train_data.txt",
"dev_data": null,
"test_data": "core/data/test_data.txt",

"num_top_label": 5,
"num_top_word": 5,
"num_top_sent": 3,
```

학습 명령어 예시

```
export CUDA_VISIBLE_DEVICES=0
python xdc_train.py \
    --model_name='nsmc_sentiment' \
    --num_train_epochs=5.0
    --train
```

4. XDC Inference

모델 경로 등 Arguments를 정확히 설정하여 inference 명령어 실행

*실행 위치 : 기본 위치

Arguments 설정 예시(models/xdc_config.json)

```
"num_top_label": 5,
"num_top_word": 5,
"num_top_sent": 3,

"xdc_checkpoint": "/models/nsmc_sentiment/"
```

실행 명령어 예시

```
export CUDA_VISIBLE_DEVICES=0
python xdc_inference.py \
    --lang='ko'
```

관련 자료


Smart City
AI Contact Center
AI English Education
Smart Factory
AI Robot

maum Academy
AI Professional Education

관련 자료

- https://github.com/google-research/bert
- https://arxiv.org/abs/1810.04805
- https://jalammar.github.io/illustrated-bert/
- https://github.com/e9t/nsmc

감사합니다.

MINDs Lab. AI Platform Company