

Level 2. How to customize AI engines



<http://mindslab.ai>

1. BERT MRC 구조

1. MRC Train/Test

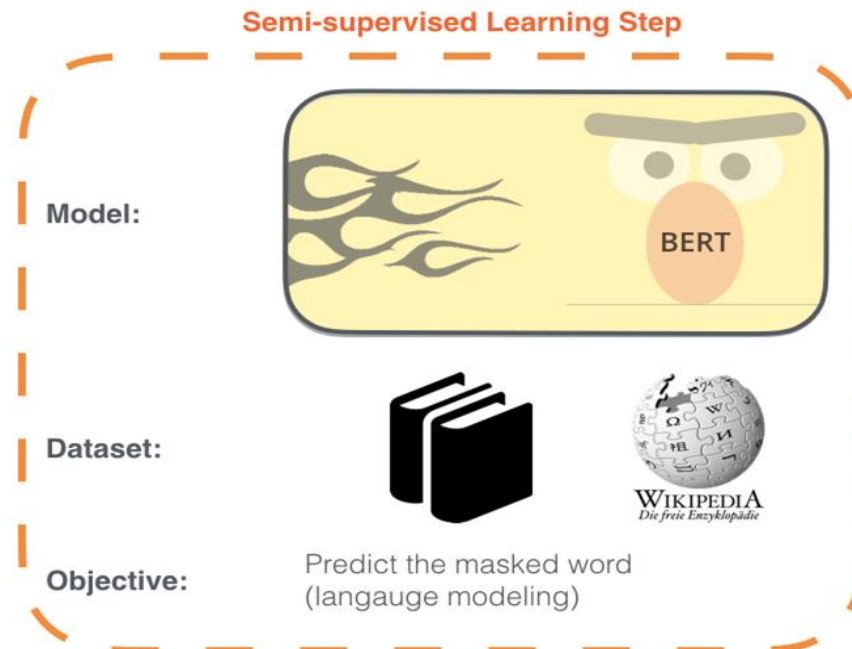
1. MRC Inference

1. MRC 실습



BERT MRC 란

- Google에서 발표한 Transformer encoder 기반의 구조를 가진 사전학습 모델을 base로 학습한 Machine Reading Comprehension 모델
- Autoencoder 방식의 사전학습 방법을 사용하여 BERT 모델 학습
(대량의 텍스트 데이터를 MLM(Masked Language Model), NSP(Next Sentence Prediction) 방식을 이용해 사전학습)
- 사전학습 된 모델에 MRC Task에 맞는 네트워크를 추가하여 학습



학습 모델

- BERT MRC 아키텍처

vocab_size: 사전 수

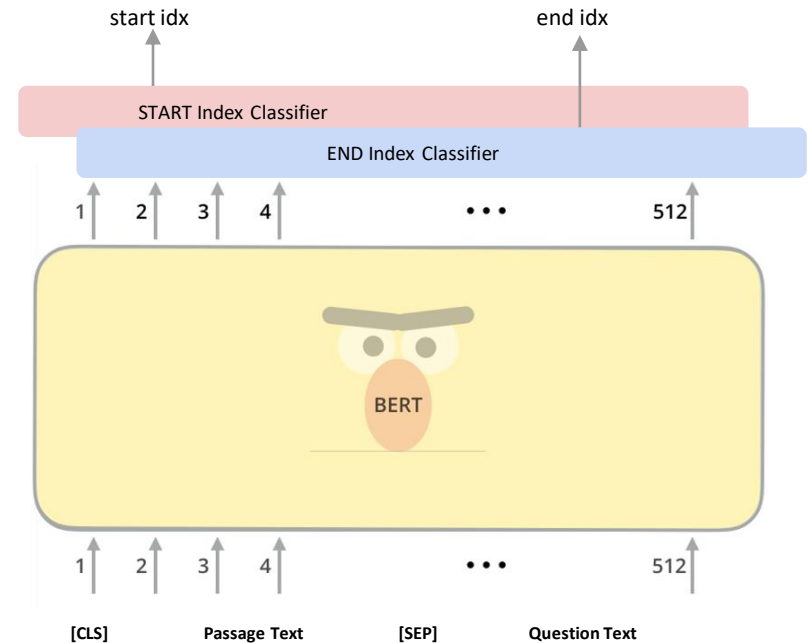
max_position_embedding: 최대 시퀀스

num_attention_heads: 어텐션 헤드 수

num_hidden_layers: 트랜스포머 블록 수

hidden_size: 임베딩 크기

```
"vocab_size": 119547
"max_position_embeddings": 512,
"num_attention_heads": 12,
"num_hidden_layers": 12,
"hidden_size": 768,
```



BERT MRC source

BERT MRC source tree

Source tree

- core: mrc modules
 - └ additional_layers.py: mrc layers
 - └ evaluate_v1.py: mrc1.0 test metric
 - └ modeling.py: BERT modules
 - └ mrc_core.py: MRC modules
 - └ tokenization.py: BERT tokenizer
 - └ optimization.py: BERT optimizer
 - └ transform_data.py: convert input format
- models: 사전학습 모델 및 학습 모델
- mrc_client.py: serving client
- mrc_server.py: serving
- mrc_runner.sh: mrc 실행 shell

핵심 Command

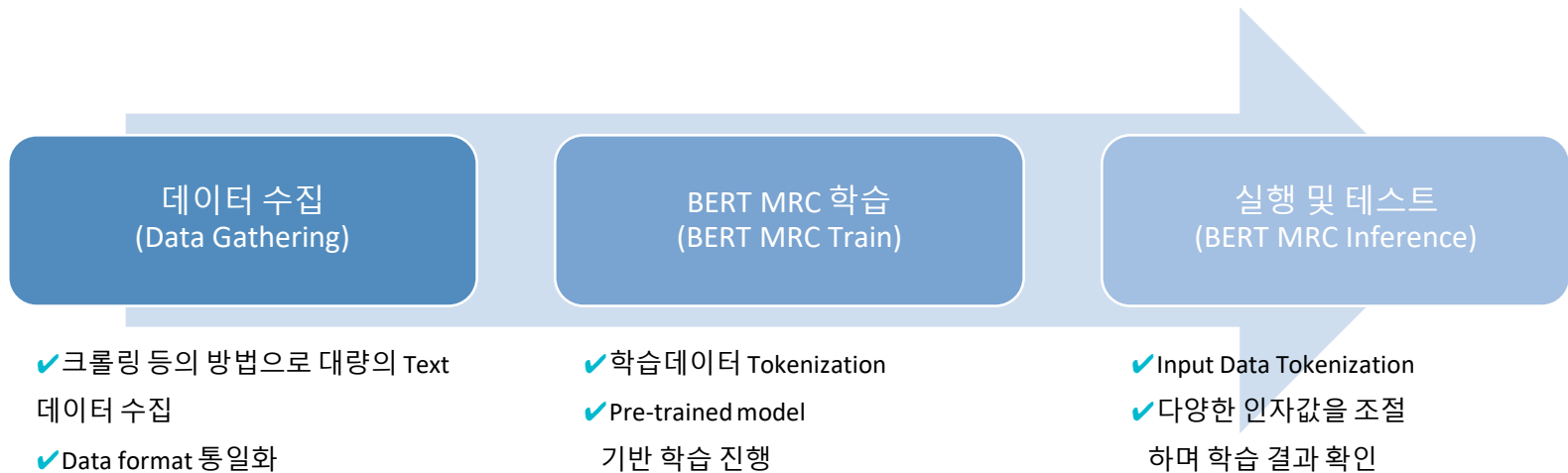
- mrc_train.py: MRC 학습 명령어
- mrc_inference.py : MRC 실행 명령어
(학습한 모델 테스트)

```
.
├── core
│   ├── additional_layers.py
│   ├── evaluate_v1.py
│   ├── __init__.py
│   ├── modeling.py
│   ├── mrc_core.py
│   ├── optimization.py
│   ├── tokenization.py
│   └── transform_data.py
├── __init__.py
├── models
│   ├── ko_bertmrc
│   └── mrc_config.json
├── mrc_client.py
├── mrc_inference.py
├── mrc_runner.sh
├── mrc_server.py
├── mrc_train.py
├── proto
│   └── mrc.proto
└── README.md
```

* MRC가 동작하기 위한 기본 서버 환경 구조

BERT MRC 학습 및 실행 Process

- 기본적으로 아래와 같이 3단계로 진행하며, 성능 안정화가 될 때까지 반복 작업



1. BERT MRC 구조

1. MRC Train/Test

1. MRC Inference

1. MRC 실습



MRC 학습 데이터

- 문서 – 질문 – 정답 쌍 데이터
- json format(필수 o)

json 구조

```
Data: [paragraphs: [context:  
                    qas: [answer: [answer_start:, text: ]  
                    id:  
                    question: ]  
]
```

* 각 list에는 다수의 값이 들어감

[참고] 데이터 파일 예시

```
{'paragraphs': [{'context': '1989년 2월 15일 여의도 농민 폭력 시위를 주도한 혐의(폭력행위등처벌에관한법률위반)으로 지명수배되었다. 1989년  
'qas': [{'answers': [{'answer_start': 0, 'text': '1989년 2월 15일'}],  
'id': '6548850-0-0',  
'question': '임종석이 여의도 농민 폭력 시위를 주도한 혐의로 지명수배 된 날은?'},  
{'answers': [{'answer_start': 125, 'text': '임수경'}],  
'id': '6548850-0-1',  
'question': '1989년 6월 30일 평양축전에 대표로 파견 된 인물은?'},
```


데이터 전처리

- Mecab 기반 sentencepiece 사전 (models/345K에 제공)
- Bert tokenizer (tokenization.py)
 - * 학습 및 Inference 실행 시 자동으로 적용됨

```
context: 1989년 2월 15일 여의도 농민 폭력 시위를 주도한 혐의(폭력행위등처벌에관한법률위반)으로 지명수배되었다. 1989년  
question: 임종석이 여의도 농민 폭력 시위를 주도한 혐의로 지명수배 된 날은?  
answers [{ 'text': '1989년 2월 15일', 'answer_start': 0}]
```

```
context: 알렉산더 메이그스 헤이그 2세(영어: Alexander Meigs Haig, Jr., 1924년 12월 2일 ~ 2010년 2월 20일)는 미국의  
question: 미국 군대 내 두번째로 높은 직위는 무엇인가?  
answers [{ 'text': '미국 육군 부참모 총장', 'answer_start': 204}]
```

mrc_train.py 명령 옵션 (1/2)

```
parser.add_argument('--model_name', type=str, required=True)
parser.add_argument('-b', '--train_batch_size', type=int, default=8)
parser.add_argument('-l', '--max_seq_length', type=int, default=384)
parser.add_argument('-i', '--num_train_epochs', type=float, default=2.0)
```

- model_name: 학습 시키려는 모델 이름
- train_batch_size: 학습 시 minibatch 크기
- max_seq_length: 모델이 읽게 되는 데이터의 최대 길이(token 기준)

(batch 와 length는 GPU ram 크기에 맞게 설정,
1080ti GPU 기준 length 384 일 때 batch_size 6으로 학습 가능)
- num_train_epochs: 학습 진행 횟수

mrc_train.py 명령 옵션 (2/2)

```
parser.add_argument('-t', '--train', action='store_true', default=False)
parser.add_argument('-e', '--en', action='store_true', default=False)
parser.add_argument('-m', '--model_type', type=int, default=3)
parser.add_argument('--train_file', type=str, default=None)
parser.add_argument('--dev_file', type=str, default=None)
```

- train: 학습 시 옵션 표시
- en: 데이터 언어 설정으로 영어인 경우 옵션 표시
- num_train_epochs: 학습 진행 횟수
- train_file: 학습 데이터 경로 입력
- dev_file: 테스트 데이터 경로 입력

영어 데이터 학습 command 예시

```
python mrc_train.py \
--model_name {모델 이름} \
--train_batch_size 6 \
--max_seq_length 384 \
--num_train_epochs 2.0 \
--train \
--en \
--train_file {학습 데이터 경로}
```

```
INFO:tensorflow:*** Features ***
INFO:tensorflow: name = end_positions, shape = (8,)
INFO:tensorflow: name = input_ids, shape = (8, 384)
INFO:tensorflow: name = input_mask, shape = (8, 384)
INFO:tensorflow: name = segment_ids, shape = (8, 384)
INFO:tensorflow: name = start_positions, shape = (8,)
INFO:tensorflow: name = unique_ids, shape = (8,)
WARNING:tensorflow:Efficient allreduce is not supported for IndexedSlices.
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:batch_all_reduce invoked for batches size = 1 with algorithm = nccl, num_packs = 1, agg_small_grads_max_bytes = 0 and
d agg_small_grads_max_group = 10
INFO:tensorflow:Done calling model fn.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
2019-11-20 17:32:12.985584: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding visible gpu devices: 0
2019-11-20 17:32:12.985698: I tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device interconnect StreamExecutor with strength
1 edge matrix:
2019-11-20 17:32:12.985718: I tensorflow/core/common_runtime/gpu/gpu_device.cc:988] 0
2019-11-20 17:32:12.985732: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0: N
2019-11-20 17:32:13.068188: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/repli
ca:0/task:0/device:GPU:0 with 10421 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:14:00.0, comp
ute capability: 6.1)
INFO:tensorflow:Running local_init op.
INFO:tensorflow:Done running local_init op.
INFO:tensorflow:Saving checkpoints for 0 into ./models/ex/model.ckpt.
INFO:tensorflow:loss = 1.0880504, step = 0
INFO:tensorflow:global_step/sec: 2.6904
INFO:tensorflow:loss = 2.0219004, step = 200 (67.792 sec)
```

MRC Test

Run MRC Test

한글 데이터 테스트 command 예시

```
python mrc_train.py \  
--model_name {모델 이름} \  
--train_batch_size 6 \  
--max_seq_length 384 \  
--num_train_epochs 2.0 \  
--dev_file {테스트 데이터 경로}
```

```
INFO:tensorflow:***** Running predictions *****  
INFO:tensorflow: Num orig examples = 30182  
INFO:tensorflow: Num split examples = 94159  
WARNING:tensorflow:From /home/minds/.virtualenvs/venv_bert/brain-lm/bert/mrc/core/mrc_core.py:725: map_and_batch (from tensorflow.co  
ntrib.data.python.ops.batching) is deprecated and will be removed in a future version.  
Instructions for updating:  
Use 'tf.data.experimental.map_and_batch(...)'.  
INFO:tensorflow:Calling model_fn.  
INFO:tensorflow:*** Features ***  
INFO:tensorflow: name = input_ids, shape = (?, 384)  
INFO:tensorflow: name = input_mask, shape = (?, 384)  
INFO:tensorflow: name = segment_ids, shape = (?, 384)  
INFO:tensorflow: name = unique_ids, shape = (?,)  
INFO:tensorflow:Done calling model_fn.  
INFO:tensorflow:Graph was finalized.  
2019-11-20 17:46:16.439051: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding visible gpu devices: 0  
2019-11-20 17:46:16.439155: I tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device interconnect StreamExecutor with strength  
1 edge matrix:  
2019-11-20 17:46:16.439176: I tensorflow/core/common_runtime/gpu/gpu_device.cc:988] 0  
2019-11-20 17:46:16.439192: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0: N  
2019-11-20 17:46:16.881381: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/repli  
ca:0/task:0/device:GPU:0 with 10421 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:14:00.0, comp  
ute capability: 6.1)  
INFO:tensorflow:Restoring parameters from ./models/ex/model.ckpt-0  
INFO:tensorflow:Running local init op.  
INFO:tensorflow:Done running local init op.  
INFO:tensorflow:Processing example: 0  
INFO:tensorflow:Processing example: 1000  
INFO:tensorflow:Processing example: 2000  
INFO:tensorflow:Processing example: 3000  
INFO:tensorflow:Processing example: 4000
```

1. BERT MRC 구조

1. MRC Train/Test

1. MRC Inference

1. MRC 실습



mrc_inference.py 명령 옵션 (1/1)

```
parser.add_argument('--model_name', type=str, required=True)
parser.add_argument('-p', '--context', type=str, default=None)
parser.add_argument('-q', '--question', type=str, default=None)
parser.add_argument('-d', '--device', type=int, default=0)
parser.add_argument('-e', '--en', action='store_true', default=False)
```

- model_name: 학습된 모델 경로 입력
- context: 문서 내용 입력
- question: 질문 내용 입력
- device: 사용 GPU 번호
- en: 영어 모델 여부

한글 모델 Inference command 예시

```
python mrc_inference.py \
--model_name {모델 이름} \
--context {본문 내용} \
--question {질문 내용} \
--device {gpu id} \
#--en
```

```
2019-11-20 18:08:31.436121: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow
binary was not compiled to use: AVX2 FMA
2019-11-20 18:08:35.712461: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1432] Found device 0 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.582
pciBusID: 0000:14:00.0
totalMemory: 10.92GiB freeMemory: 10.77GiB
2019-11-20 18:08:35.712543: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding visible gpu devices: 0
2019-11-20 18:08:38.971485: I tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device interconnect StreamExecutor with strength
1 edge matrix:
2019-11-20 18:08:38.971551: I tensorflow/core/common_runtime/gpu/gpu_device.cc:988] 0
2019-11-20 18:08:38.971568: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0: N
2019-11-20 18:08:38.972016: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/repli
ca:0/task:0/device:GPU:0 with 10421 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:14:00.0, comp
ute capability: 6.1)
['김영관 법 (부정청탁 및 금품수수 금지법) 시행 이후 많은 첫 주말인 지난 1일 오후 6시쯤, 기자는 서울 서초구 G금익신고 학원을 찾아갔다. 김
영관 법 위반자를 적발해 표창을 받는 방법을 알려 주는 일명 관파라치 (김영관 법+파라치) 양성 학원이다. 교육 수강생 이모 (56자영업)씨와
박모 (57자직자)씨가 다소 긴장한 표정으로 앉아 있었다. 두 사람은 최근 이 학원에서 김영관 법 위반자를 손쉽게 탄속할 수 있는 비법 등 이른
교육 (총 3시간 30분)을 받은 수강생이다. 이들은 이날 서울의 한 병원 장례식장으로 첫 현장 출동을 앞둔 상황이었다. 이 학원의 문성숙 대표
가 카메라를 사용법과 현장 활동을 주의사항을 전달했다. 카메라라는 소재 끝 안쪽에 숨겨 촬영이 가능한 형태였다. 이 학원 강의실 한쪽 장
식장 안에는 안경부터 모자라 이터영합지갑 등 다양한 형태의 카메라가 진열돼 있었다. 학원 관계자들은 최근 일부 관파라치 학원이 수강
생들에게 카메라를 시종가 격보다 싸게 팔았다는 비판 보도를 의식한 때문인지 가격은 절저히 합구했다.']
['김영관 법이 뭐야']
OrderedDict([('text', '부정청탁 및 금품수수 금지법'), ('probability', 0.9975685906370428), ('start_index', 14), ('end_index', 25), ('
start_logit', 8.977004051208496), ('end_logit', 8.652633666992188), ('passage_idx', 0)])
```


1. BERT MRC 구조

1. MRC Train/Test

1. MRC Inference

1. MRC 실습



실습 주제

- 각자 준비한 데이터 셋을 모델에 읽어 학습하고 Inference 해보기
- 데이터 셋 전처리 작업의 경우 모델 실행시 자동으로 이루어짐
- 정해진 json format에 맞게 데이터를 준비하는 것이 중요

1. 서버 접속

- 1) Xshell, Putty 등 ssh client 프로그램을 통하여 아래 서버에 접속
- 2) MRC가 설치되어 있는 위치로 이동 (이하, '**기본 위치**'라 함)

2. 학습데이터 준비

- 1) json 형식의 학습데이터를 준비하여 원하는 위치에 옮겨놓는다.

3. MRC 학습

학습데이터 경로 등 Arguments를 정확히 설정하여 학습 명령어 실행

*실행 위치 : 기본 위치

명령어 예시

```
python mrc_train.py \  
--model_name ex \  
--train_batch_size 6 \  
--max_seq_length 384 \  
--num_train_epochs 2.0 \  
--train \  
--train_file /data/train_data.json
```

4. MRC Inference

다양한 Arguments를 정확히 설정하여 Inference 명령어 실행

*실행 위치 : 기본 위치

명령어 예시

```
python mrc_inference.py \  
--model_name ex \  
--context '안녕하세요 마인즈랩입니다.' \  
--question '누구?' \  
--device 0 \
```

감사합니다.

MINDs Lab. AI Platform Company