model_06_bert_cross_encoder_classification

May 8, 2023

1 Model 06 Bert Cross Entropy Classification for Label Prediction

Prediction of claim labels based on the matched evidence.

1.1 Setup

1.1.1 Working Directory

[ ]:

```
# Change the working directory to project root
from pathlib import Path
import os
ROOT_DIR = Path.cwd()
while not ROOT_DIR.joinpath("src").exists():
    ROOT_DIR = ROOT_DIR.parent
os.chdir(ROOT_DIR)
```

1.1.2 Dependencies

[ ]:

```
# Imports and dependencies
import torch
from torch.utils.data import DataLoader
from torch.nn import CrossEntropyLoss
from torch.optim import AdamW
from torch.optim.lr_scheduler import LinearLR
from torcheval.metrics import MulticlassAccuracy, MulticlassF1Score

from src.logger import SimpleLogger
from src.model_05 import BertCrossEncoderClassifier
from src.data import LabelClassificationDataset
from src.torch_utils import get_torch_device
import json
from dataclasses import dataclass
from typing import List, Union, Tuple
from tqdm import tqdm
import random
import numpy as np
from datetime import datetime
from math import exp
```

```python
from sklearn.model_selection import ParameterGrid

TORCH_DEVICE = get_torch_device()
```

```
/opt/homebrew/Caskroom/miniconda/base/envs/comp90042_project/lib/python3.8/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

```
Torch device is 'mps'
```

1.1.3 File paths

[ ]:

```python
MODEL_PATH = ROOT_DIR.joinpath("./result/models/*")
DATA_PATH = ROOT_DIR.joinpath("./data/*")
LOG_PATH = ROOT_DIR.joinpath("./result/logs/*")
SHORTLIST_PATH = ROOT_DIR.joinpath("./result/pipeline/shortlisting_v2/*")

run_time = datetime.now().strftime('%Y_%m_%d_%H_%M')
```

1.2 Training Loop

[ ]:

```python
def training_loop(
    model,
    claims_paths:List[Path],
    save_path:Path=None,
    warmup:float=0.1,
    lr:float=0.00005, # 5e-5
    weight_decay:float=0.01,
    normalize_text:bool=True,
    max_length:int=128,
    dropout:float=None,
    n_epochs:int=5,
    batch_size:int=64,
):
    # Generate training dataset
    train_data = LabelClassificationDataset(
        claims_paths=claims_paths,
        training=True,
    )
    train_dataloader = DataLoader(
        dataset=train_data,
        shuffle=True,
        batch_size=batch_size
    )
    # Generate evaluation dataset
    dev_data = LabelClassificationDataset(
        claims_paths=[Path("./data/dev-claims.json")],
        training=True,
    )
    dev_dataloader = DataLoader(
        dataset=dev_data,
        shuffle=False,
        batch_size=batch_size
    )

    # Loss function
    loss_fn = CrossEntropyLoss()

    # Optimizer
    optimizer = AdamW(
        params=model.parameters(),
        lr=lr,
        weight_decay=weight_decay
    )

    # Scheduler
    scheduler = LinearLR(
        optimizer=optimizer,
        total_iters=warmup * len(train_dataloader),
        verbose=False
    )

    # Metrics
    accuracy_fn = MulticlassAccuracy()
    f1_fn = MulticlassF1Score()

    # Training epochs --------------------------------------------------------

    best_epoch_loss = 999
    best_epoch_f1 = -1
    best_epoch_acc = -1
    best_epoch = 0
    for epoch in range(n_epochs):

        print(f"Epoch: {epoch + 1} of {n_epochs}\n")

        # Run training -------------------------------------------------------
        model.train()

        train_batches = tqdm(train_dataloader, desc="train batches")
        running_losses = []
        for batch in train_batches:
            claim_texts, evidence_texts, labels, claim_ids, evidence_ids = batch
            texts = list(zip(claim_texts, evidence_texts))

            # Reset optimizer
            optimizer.zero_grad()

            # Forward + loss
            output, logits, seq = model(
                texts=texts,
                normalize_text=normalize_text,
                max_length=max_length,
                dropout=dropout
            )
            loss = loss_fn(logits, labels)

            # Backward + optimizer
            loss.backward()
            optimizer.step()

            # Update running loss
            batch_loss = loss.item() * len(batch)
            running_losses.append(batch_loss)

            train_batches.postfix = f"loss: {batch_loss:.3f}"

            # Update scheduler
            scheduler.step()

            continue

        # Epoch loss
        epoch_loss = np.average(running_losses)
        print(f"Average epoch loss: {epoch_loss:.3f}")

        # Run evaluation ------------------------------------------------------
        model.eval()

        dev_batches = tqdm(dev_dataloader, desc="dev batches")
        dev_acc = []
        dev_f1 = []
        for batch in dev_batches:
            claim_texts, evidence_texts, labels, claim_ids, evidence_ids = batch
            texts = list(zip(claim_texts, evidence_texts))

            # Forward
            output, logits, seq = model(
                texts=texts,
                normalize_text=normalize_text,
                max_length=max_length,
                dropout=dropout
            )

            # Prediction
            predicted = torch.argmax(output, dim=1)

            # Metrics
            accuracy_fn.update(predicted.cpu(), labels.cpu())
            f1_fn.update(predicted.cpu(), labels.cpu())

            acc = accuracy_fn.compute()
            f1 = f1_fn.compute()

            dev_acc.append(acc)
            dev_f1.append(f1)

            dev_batches.postfix = f" acc: {acc:.3f}, f1: {f1:.3f}"

            continue

        # Consider metrics
        epoch_acc = np.average(dev_acc)
        print(f"Average epoch accuracy: {epoch_acc:.3f}")

        epoch_f1 = np.average(dev_f1)
        print(f"Average epoch f1: {epoch_f1:.3f}")

        if epoch_acc > best_epoch_acc:
            best_epoch_acc = epoch_acc

        if epoch_f1 > best_epoch_f1:
            best_epoch_f1 = epoch_f1
            best_epoch = epoch + 1

        # Save model ----------------------------------------------------------

        # Save the model with the best f1 score
        if save_path and epoch_f1 >= best_epoch_f1:
            torch.save(model, save_path)
            print(f"Saved model to: {save_path}")

    print("Done!")
    return best_epoch_acc, best_epoch_f1, best_epoch
```

1.3 Load model

Use a blank pre-trained

[ ]:

```
model = BertCrossEncoderClassifier(
    pretrained_name="bert-base-uncased",
    n_classes=3,
    device=TORCH_DEVICE
)
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
```

Or load one previously trained

[ ]:

```
# MODEL_SAVE_PATH = MODEL_PATH.with_name("")
# with open(MODEL_PATH.with_name(MODEL_SAVE_PATH), mode="rb") as f:
#     model = torch.load(f, map_location=TORCH_DEVICE)
```

1.4 Training and evaluation loop

[ ]:

```
# training_loop(
#     model=model,
#     claims_paths=[
#         DATA_PATH.with_name("train-claims.json")
#     ],
#     save_path=MODEL_PATH.
 ↪with_name(f"model_06_bert_cross_encoder_label_{run_time}.pth"),
#     warmup=0.1,
#     lr=0.00005, # 5e-5
#     weight_decay=0.01,
#     normalize_text=True,
#     max_length=512,
#     dropout=None,
#     n_epochs=1,
#     batch_size=24,
# )
```

1.5 Tune hyperparameters

[ ]:

```
hyperparams = ParameterGrid(param_grid={
    "claims_paths": [[
        DATA_PATH.with_name("train-claims.json")
    ]],
    "warmup": [0.1],
    "lr": [0.000005],
    "weight_decay": [0.01, 0.02],
    "normalize_text": [True],
    "max_length": [512],
    "dropout": [None, 0.1],
    "n_epochs": [5],
    "batch_size": [24],
    "freeze_bert": [False, True]
})
```

[ ]:

```
import warnings
warnings.filterwarnings('ignore')
```

[ ]:

```
with SimpleLogger("model_06_cross_encoder_retrieval") as logger:
    logger.set_stream_handler()
    logger.set_file_handler(
        log_path=LOG_PATH,
        filename="model_06_hyperparam_tuning.txt"
    )
    best_f1 = -1
    best_params = {}
    for hyperparam in hyperparams:
        model = BertCrossEncoderClassifier(
            pretrained_name="bert-base-uncased",
            n_classes=3,
            device=TORCH_DEVICE
        )

        model_param = hyperparam

        # Freeze bert parameters if desired
        if "freeze_bert" in model_param.keys():
            if hyperparam["freeze_bert"] is True:
                for param in model.bert.parameters():
                    param.requires_grad = False
            del model_param["freeze_bert"]

        logger.info("\n== RUN")
        logger.info(hyperparam)
        accuracy, f1, epoch = training_loop(model=model, **model_param)

        logger.info(f"run_best_epoch: {epoch}, run_best_acc: {accuracy},␣
↪run_best_f1: {f1}")

        if f1 > best_f1:
            best_f1 = f1
            best_params = hyperparam

        logger.info(f"\n== CURRENT BEST F1: {best_f1}")
        logger.info(best_params)
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
```

```
2023-05-08 10:42:43 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 10:42:43 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'
```

```
claims: 100%|      | 1228/1228 [00:00<00:00, 451332.40it/s]
```

```
generated dataset n=3730
Torch device is 'mps'
```

```
claims: 100%|      | 154/154 [00:00<00:00, 446079.29it/s]
```

```
generated dataset n=433
Epoch: 1 of 5
```

```
train batches: 100%|      | 156/156 [03:22<00:00,  1.30s/it, loss: 5.104]
```

```
Average epoch loss: 5.061

dev batches: 100%|     | 19/19 [00:07<00:00,  2.63it/s,  acc: 0.513, f1:
0.513]

Average epoch accuracy: 0.527
Average epoch f1: 0.527
Epoch: 2 of 5


train batches: 100%|     | 156/156 [03:22<00:00,  1.30s/it, loss: 5.747]

Average epoch loss: 4.273

dev batches: 100%|     | 19/19 [00:07<00:00,  2.67it/s,  acc: 0.543, f1:
0.543]

Average epoch accuracy: 0.540
Average epoch f1: 0.540
Epoch: 3 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.30s/it, loss: 1.703]

Average epoch loss: 3.054

dev batches: 100%|     | 19/19 [00:07<00:00,  2.69it/s,  acc: 0.557, f1:
0.557]

Average epoch accuracy: 0.554
Average epoch f1: 0.554
Epoch: 4 of 5


train batches: 100%|     | 156/156 [03:22<00:00,  1.30s/it, loss: 1.008]

Average epoch loss: 1.810

dev batches: 100%|     | 19/19 [00:07<00:00,  2.69it/s,  acc: 0.554, f1:
0.554]

Average epoch accuracy: 0.558
Average epoch f1: 0.558
Epoch: 5 of 5


train batches: 100%|     | 156/156 [03:21<00:00,  1.29s/it, loss: 0.530]

Average epoch loss: 1.178

dev batches: 100%|     | 19/19 [00:07<00:00,  2.69it/s,  acc: 0.552, f1:
0.552]

Average epoch accuracy: 0.554
Average epoch f1: 0.554
Done!
2023-05-08 11:00:14 model_06_cross_encoder_retrieval:INFO
```

```
run_best_epoch: 4, run_best_acc: 0.5582168698310852, run_best_f1:
0.5582168698310852

2023-05-08 11:00:14 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.5582168698310852

2023-05-08 11:00:14 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.01}



Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
2023-05-08 11:00:16 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 11:00:16 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'
claims: 100%|     | 1228/1228 [00:00<00:00, 608243.42it/s]
generated dataset n=3730
Torch device is 'mps'
claims: 100%|     | 154/154 [00:00<00:00, 599186.29it/s]
generated dataset n=433
Epoch: 1 of 5
```

```
train batches: 100%|     | 156/156 [03:23<00:00,  1.30s/it, loss: 5.348]

Average epoch loss: 4.626

dev batches: 100%|     | 19/19 [00:07<00:00,  2.68it/s,  acc: 0.543, f1:
0.543]

Average epoch accuracy: 0.595
Average epoch f1: 0.595
Epoch: 2 of 5


train batches: 100%|     | 156/156 [03:22<00:00,  1.30s/it, loss: 3.716]

Average epoch loss: 3.771

dev batches: 100%|     | 19/19 [00:07<00:00,  2.69it/s,  acc: 0.553, f1:
0.553]

Average epoch accuracy: 0.563
Average epoch f1: 0.563
Epoch: 3 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.30s/it, loss: 2.130]

Average epoch loss: 3.017

dev batches: 100%|     | 19/19 [00:07<00:00,  2.68it/s,  acc: 0.549, f1:
0.549]

Average epoch accuracy: 0.557
Average epoch f1: 0.557
Epoch: 4 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.30s/it, loss: 0.993]

Average epoch loss: 2.298

dev batches: 100%|     | 19/19 [00:07<00:00,  2.70it/s,  acc: 0.550, f1:
0.550]

Average epoch accuracy: 0.555
Average epoch f1: 0.555
Epoch: 5 of 5


train batches: 100%|     | 156/156 [03:25<00:00,  1.32s/it, loss: 0.795]

Average epoch loss: 1.793

dev batches: 100%|     | 19/19 [00:07<00:00,  2.66it/s,  acc: 0.554, f1:
0.554]

Average epoch accuracy: 0.557
Average epoch f1: 0.557
```

```
Done!
2023-05-08 11:17:51 model_06_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.5950567126274109, run_best_f1:
0.5950567126274109

2023-05-08 11:17:51 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.5950567126274109

2023-05-08 11:17:51 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.02}



Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
2023-05-08 11:17:53 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 11:17:53 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'
claims: 100%|      | 1228/1228 [00:00<00:00, 586228.69it/s]
generated dataset n=3730
Torch device is 'mps'
claims: 100%|      | 154/154 [00:00<00:00, 544164.12it/s]
generated dataset n=433
```

```
Epoch: 1 of 5


train batches: 100%|     | 156/156 [01:21<00:00,  1.91it/s, loss: 5.254]

Average epoch loss: 5.788

dev batches: 100%|     | 19/19 [00:07<00:00,  2.61it/s,  acc: 0.395, f1:
0.395]

Average epoch accuracy: 0.407
Average epoch f1: 0.407
Epoch: 2 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 4.478]

Average epoch loss: 5.176

dev batches: 100%|     | 19/19 [00:07<00:00,  2.63it/s,  acc: 0.431, f1:
0.431]

Average epoch accuracy: 0.418
Average epoch f1: 0.418
Epoch: 3 of 5


train batches: 100%|     | 156/156 [01:21<00:00,  1.92it/s, loss: 4.765]

Average epoch loss: 4.997

dev batches: 100%|     | 19/19 [00:07<00:00,  2.59it/s,  acc: 0.444, f1:
0.444]

Average epoch accuracy: 0.438
Average epoch f1: 0.438
Epoch: 4 of 5


train batches: 100%|     | 156/156 [01:21<00:00,  1.91it/s, loss: 5.408]

Average epoch loss: 4.944

dev batches: 100%|     | 19/19 [00:07<00:00,  2.61it/s,  acc: 0.451, f1:
0.451]

Average epoch accuracy: 0.448
Average epoch f1: 0.448
Epoch: 5 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.94it/s, loss: 5.435]

Average epoch loss: 4.921

dev batches: 100%|     | 19/19 [00:07<00:00,  2.65it/s,  acc: 0.455, f1:
0.455]
```

```
Average epoch accuracy: 0.453
Average epoch f1: 0.453
Done!
2023-05-08 11:25:17 model_06_cross_encoder_retrieval:INFO
run_best_epoch: 5, run_best_acc: 0.4528111517429352, run_best_f1:
0.4528111517429352

2023-05-08 11:25:17 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.5950567126274109

2023-05-08 11:25:17 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.02}


Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).

2023-05-08 11:25:18 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 11:25:18 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 445746.89it/s]

generated dataset n=3730
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 529444.93it/s]
```

```
generated dataset n=433
Epoch: 1 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 5.390]

Average epoch loss: 5.544

dev batches: 100%|     | 19/19 [00:07<00:00,  2.50it/s,  acc: 0.469, f1:
0.469]

Average epoch accuracy: 0.470
Average epoch f1: 0.470
Epoch: 2 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.94it/s, loss: 4.296]

Average epoch loss: 5.118

dev batches: 100%|     | 19/19 [00:07<00:00,  2.58it/s,  acc: 0.468, f1:
0.468]

Average epoch accuracy: 0.466
Average epoch f1: 0.466
Epoch: 3 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.94it/s, loss: 5.576]

Average epoch loss: 5.014

dev batches: 100%|     | 19/19 [00:07<00:00,  2.65it/s,  acc: 0.467, f1:
0.467]

Average epoch accuracy: 0.466
Average epoch f1: 0.466
Epoch: 4 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.94it/s, loss: 4.160]

Average epoch loss: 4.973

dev batches: 100%|     | 19/19 [00:07<00:00,  2.64it/s,  acc: 0.468, f1:
0.468]

Average epoch accuracy: 0.467
Average epoch f1: 0.467
Epoch: 5 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.94it/s, loss: 4.642]

Average epoch loss: 4.963
```

```
dev batches: 100%|      | 19/19 [00:07<00:00,  2.65it/s,  acc: 0.468, f1:
0.468]

Average epoch accuracy: 0.467
Average epoch f1: 0.467
Done!
2023-05-08 11:32:39 model_06_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.46990710496902466, run_best_f1:
0.46990710496902466

2023-05-08 11:32:39 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.5950567126274109

2023-05-08 11:32:39 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.02}


Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).

2023-05-08 11:32:41 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 11:32:41 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': 0.1, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5, 'normalize_text':
True, 'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|      | 1228/1228 [00:00<00:00, 737064.30it/s]

generated dataset n=3730
```

```
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 608786.82it/s]

generated dataset n=433
Epoch: 1 of 5


train batches: 100%|     | 156/156 [03:24<00:00,  1.31s/it, loss: 4.301]

Average epoch loss: 4.702

dev batches: 100%|     | 19/19 [00:07<00:00,  2.63it/s,  acc: 0.490, f1:
0.490]

Average epoch accuracy: 0.560
Average epoch f1: 0.560
Epoch: 2 of 5


train batches: 100%|     | 156/156 [03:24<00:00,  1.31s/it, loss: 4.390]

Average epoch loss: 3.891

dev batches: 100%|     | 19/19 [00:07<00:00,  2.68it/s,  acc: 0.501, f1:
0.501]

Average epoch accuracy: 0.514
Average epoch f1: 0.514
Epoch: 3 of 5


train batches: 100%|     | 156/156 [03:24<00:00,  1.31s/it, loss: 3.254]

Average epoch loss: 3.096

dev batches: 100%|     | 19/19 [00:07<00:00,  2.66it/s,  acc: 0.505, f1:
0.505]

Average epoch accuracy: 0.511
Average epoch f1: 0.511
Epoch: 4 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.31s/it, loss: 1.678]

Average epoch loss: 2.364

dev batches: 100%|     | 19/19 [00:07<00:00,  2.64it/s,  acc: 0.525, f1:
0.525]

Average epoch accuracy: 0.524
Average epoch f1: 0.524
Epoch: 5 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.31s/it, loss: 0.803]
```

```
Average epoch loss: 1.682

dev batches: 100%|      | 19/19 [00:07<00:00,  2.59it/s,  acc: 0.536, f1:
0.536]

Average epoch accuracy: 0.535
Average epoch f1: 0.535
Done!
2023-05-08 11:50:19 model_06_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.5603758096694946, run_best_f1:
0.5603758096694946

2023-05-08 11:50:19 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.5950567126274109

2023-05-08 11:50:19 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': None, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5,
'normalize_text': True, 'warmup': 0.1, 'weight_decay': 0.02}



Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).

2023-05-08 11:50:21 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 11:50:21 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': 0.1, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5, 'normalize_text':
True, 'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'
```

```
claims: 100%|     | 1228/1228 [00:00<00:00, 594964.23it/s]

generated dataset n=3730
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 544164.12it/s]

generated dataset n=433
Epoch: 1 of 5


train batches: 100%|     | 156/156 [03:24<00:00,  1.31s/it, loss: 3.872]

Average epoch loss: 4.627

dev batches: 100%|     | 19/19 [00:07<00:00,  2.69it/s,  acc: 0.605, f1:
0.605]

Average epoch accuracy: 0.633
Average epoch f1: 0.633
Epoch: 2 of 5


train batches: 100%|     | 156/156 [03:24<00:00,  1.31s/it, loss: 4.911]

Average epoch loss: 3.832

dev batches: 100%|     | 19/19 [00:07<00:00,  2.54it/s,  acc: 0.584, f1:
0.584]

Average epoch accuracy: 0.602
Average epoch f1: 0.602
Epoch: 3 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.31s/it, loss: 2.990]

Average epoch loss: 3.053

dev batches: 100%|     | 19/19 [00:07<00:00,  2.53it/s,  acc: 0.572, f1:
0.572]

Average epoch accuracy: 0.583
Average epoch f1: 0.583
Epoch: 4 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.31s/it, loss: 2.770]

Average epoch loss: 2.258

dev batches: 100%|     | 19/19 [00:07<00:00,  2.63it/s,  acc: 0.574, f1:
0.574]

Average epoch accuracy: 0.576
Average epoch f1: 0.576
```

```
Epoch: 5 of 5


train batches: 100%|     | 156/156 [03:23<00:00,  1.30s/it, loss: 1.144]

Average epoch loss: 1.575

dev batches: 100%|     | 19/19 [00:07<00:00,  2.63it/s,  acc: 0.566, f1:
0.566]

Average epoch accuracy: 0.573
Average epoch f1: 0.573
Done!
2023-05-08 12:07:58 model_06_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.6332693696022034, run_best_f1:
0.6332693696022034

2023-05-08 12:07:58 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.6332693696022034

2023-05-08 12:07:58 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': 0.1, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5, 'normalize_text':
True, 'warmup': 0.1, 'weight_decay': 0.02}



Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).

2023-05-08 12:08:00 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 12:08:00 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': 0.1, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5, 'normalize_text':
```

```
True, 'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 559909.26it/s]

generated dataset n=3730
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 540521.18it/s]

generated dataset n=433
Epoch: 1 of 5


train batches: 100%|     | 156/156 [01:21<00:00,  1.91it/s, loss: 5.503]

Average epoch loss: 6.136

dev batches: 100%|     | 19/19 [00:07<00:00,  2.62it/s,  acc: 0.372, f1:
0.372]

Average epoch accuracy: 0.389
Average epoch f1: 0.389
Epoch: 2 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.94it/s, loss: 4.714]

Average epoch loss: 5.319

dev batches: 100%|     | 19/19 [00:07<00:00,  2.64it/s,  acc: 0.419, f1:
0.419]

Average epoch accuracy: 0.400
Average epoch f1: 0.400
Epoch: 3 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 4.595]

Average epoch loss: 5.056

dev batches: 100%|     | 19/19 [00:07<00:00,  2.55it/s,  acc: 0.437, f1:
0.437]

Average epoch accuracy: 0.432
Average epoch f1: 0.432
Epoch: 4 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 4.130]

Average epoch loss: 4.960

dev batches: 100%|     | 19/19 [00:07<00:00,  2.65it/s,  acc: 0.447, f1:
0.447]
```

```
Average epoch accuracy: 0.444
Average epoch f1: 0.444
Epoch: 5 of 5


train batches: 100%|     | 156/156 [01:21<00:00,  1.92it/s, loss: 4.696]

Average epoch loss: 4.931

dev batches: 100%|     | 19/19 [00:07<00:00,  2.59it/s,  acc: 0.448, f1:
0.448]

Average epoch accuracy: 0.448
Average epoch f1: 0.448
Done!
2023-05-08 12:15:22 model_06_cross_encoder_retrieval:INFO
run_best_epoch: 5, run_best_acc: 0.4476720988750458, run_best_f1:
0.4476720988750458

2023-05-08 12:15:22 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.6332693696022034

2023-05-08 12:15:22 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': 0.1, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5, 'normalize_text':
True, 'warmup': 0.1, 'weight_decay': 0.02}


Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias',
'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.bias',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight',
'cls.predictions.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).

2023-05-08 12:15:24 model_06_cross_encoder_retrieval:INFO

== RUN

2023-05-08 12:15:24 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': 0.1, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5, 'normalize_text':
True, 'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 764071.40it/s]

generated dataset n=3730
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 330564.39it/s]

generated dataset n=433
Epoch: 1 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 5.045]

Average epoch loss: 4.928

dev batches: 100%|     | 19/19 [00:07<00:00,  2.59it/s,  acc: 0.455, f1:
0.455]

Average epoch accuracy: 0.480
Average epoch f1: 0.480
Epoch: 2 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 4.499]

Average epoch loss: 4.900

dev batches: 100%|     | 19/19 [00:07<00:00,  2.67it/s,  acc: 0.471, f1:
0.471]

Average epoch accuracy: 0.462
Average epoch f1: 0.462
Epoch: 3 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 5.631]

Average epoch loss: 4.894

dev batches: 100%|     | 19/19 [00:07<00:00,  2.66it/s,  acc: 0.470, f1:
0.470]

Average epoch accuracy: 0.470
Average epoch f1: 0.470
Epoch: 4 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.93it/s, loss: 4.576]

Average epoch loss: 4.878
```

```
dev batches: 100%|     | 19/19 [00:07<00:00,  2.66it/s,  acc: 0.473, f1:
0.473]

Average epoch accuracy: 0.472
Average epoch f1: 0.472
Epoch: 5 of 5


train batches: 100%|     | 156/156 [01:20<00:00,  1.94it/s, loss: 4.776]

Average epoch loss: 4.892

dev batches: 100%|     | 19/19 [00:07<00:00,  2.53it/s,  acc: 0.474, f1:
0.474]

Average epoch accuracy: 0.475
Average epoch f1: 0.475
Done!
2023-05-08 12:22:45 model_06_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.4796091318130493, run_best_f1:
0.4796091318130493

2023-05-08 12:22:45 model_06_cross_encoder_retrieval:INFO

== CURRENT BEST F1: 0.6332693696022034

2023-05-08 12:22:45 model_06_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'dropout': 0.1, 'lr': 5e-06, 'max_length': 512, 'n_epochs': 5, 'normalize_text':
True, 'warmup': 0.1, 'weight_decay': 0.02}
```