# model_05_bert_cross_encoder_retrieval_classifier_grid_search

May 8, 2023

## 1 Model 05 Bert Cross Entropy Classification for Retrieval

### 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 BinaryAccuracy, BinaryF1Score

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 sklearn.model_selection import ParameterGrid

from src.model_05 import BertCrossEncoderClassifier
from src.data import RetrievalWithShortlistDataset, RetrievalDevEvalDataset
from src.logger import SimpleLogger
```

```
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

[ ]:
```
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

[ ]:
```
def training_loop(
    model,
    claims_paths:List[Path],
    claims_shortlist_paths:List[Path],
    save_path:Path=None,
    n_neg_samples:int=5,
    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 = RetrievalWithShortlistDataset(
        claims_paths=claims_paths,
        claims_shortlist_paths=claims_shortlist_paths,
        n_neg_samples=n_neg_samples,
        pos_label=1,
        neg_label=0
    )
    train_dataloader = DataLoader(
        dataset=train_data,
        shuffle=True,
```

```
    batch_size=batch_size
)

# Generate evaluation dataset
dev_data = RetrievalDevEvalDataset(
    n_neg_samples=n_neg_samples,
    pos_label=1,
    neg_label=0,
)
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 = BinaryAccuracy()
f1_fn = BinaryF1Score()

# 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.max(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=2,
#     device=TORCH_DEVICE
# )
```

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")
#     ],
#     claims_shortlist_paths=[
#         Path("./result/pipeline/shortlisting_v2/
 ↪train_retrieved_evidences_max_500_no_rel.json"),
#     ],
#     save_path=MODEL_PATH.
 ↪with_name(f"model_05_bert_cross_encoder_retrieval_{run_time}.pth"),
#     n_neg_samples=5,
#     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")
    ]],
    "claims_shortlist_paths": [[
        Path("./result/pipeline/shortlisting_v2/
 ↪train_retrieved_evidences_max_500_no_rel.json"),
    ]],
    "n_neg_samples": [3, 5, 10],
    "warmup": [0.1],
    "lr": [0.00005, 0.0005],
    "weight_decay": [0.01, 0.02],
    "normalize_text": [True, False],
    "max_length": [512],
    "dropout": [None, 0.1],
    "n_epochs": [5, 10],
    "batch_size": [24]
})
```

[ ]:
```
with SimpleLogger("model_05_cross_encoder_retrieval") as logger:
    logger.set_stream_handler()
    logger.set_file_handler(
        log_path=LOG_PATH,
        filename="model_05_hyperparam_tuning.txt"
    )
    best_f1 = -1
    best_params = {}
    for hyperparam in hyperparams:
        model = BertCrossEncoderClassifier(
            pretrained_name="bert-base-uncased",
            n_classes=2,
            device=TORCH_DEVICE
        )
        logger.info("== RUN")
        logger.info(hyperparam)

        accuracy, f1, epoch = training_loop(model=model, **hyperparam)

        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"== 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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 13:47:41 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 13:47:41 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|      | 1228/1228 [00:00<00:00, 127332.64it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|      | 154/154 [00:00<00:00, 406240.76it/s]

generated dataset n=953
Epoch: 1 of 5


train batches: 100%|      | 326/326 [07:13<00:00,  1.33s/it, loss: 1.024]

Average epoch loss: 1.580

dev batches: 100%|      | 40/40 [00:15<00:00,  2.53it/s,  acc: 0.799, f1:
0.820]

Average epoch accuracy: 0.811
Average epoch f1: 0.834
Epoch: 2 of 5
```

```
train batches: 100%|     | 326/326 [07:08<00:00,  1.31s/it, loss: 0.116]

Average epoch loss: 0.508

dev batches: 100%|     | 40/40 [00:15<00:00,  2.58it/s,  acc: 0.687, f1:
0.754]

Average epoch accuracy: 0.728
Average epoch f1: 0.778
Epoch: 3 of 5


train batches: 100%|     | 326/326 [07:04<00:00,  1.30s/it, loss: 0.012]

Average epoch loss: 0.281

dev batches: 100%|     | 40/40 [00:15<00:00,  2.57it/s,  acc: 0.661, f1:
0.738]

Average epoch accuracy: 0.672
Average epoch f1: 0.746
Epoch: 4 of 5


train batches: 100%|     | 326/326 [07:04<00:00,  1.30s/it, loss: 0.003]

Average epoch loss: 0.182

dev batches: 100%|     | 40/40 [00:15<00:00,  2.58it/s,  acc: 0.667, f1:
0.739]

Average epoch accuracy: 0.664
Average epoch f1: 0.739
Epoch: 5 of 5


train batches: 100%|     | 326/326 [07:04<00:00,  1.30s/it, loss: 0.010]

Average epoch loss: 0.182

dev batches: 100%|     | 40/40 [00:15<00:00,  2.57it/s,  acc: 0.653, f1:
0.729]

Average epoch accuracy: 0.659
Average epoch f1: 0.733
Done!
2023-05-07 14:24:36 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.811083972454071, run_best_f1:
0.8339534997940063

2023-05-07 14:24:36 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 14:24:36 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
```

```
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 14:24:45 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 14:24:45 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 166578.44it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 328379.67it/s]

generated dataset n=953
Epoch: 1 of 5


train batches: 100%|     | 326/326 [07:06<00:00,  1.31s/it, loss: 1.680]

Average epoch loss: 1.596

dev batches: 100%|     | 40/40 [00:15<00:00,  2.57it/s,  acc: 0.665, f1:
0.727]
```

```
Average epoch accuracy: 0.673
Average epoch f1: 0.736
Epoch: 2 of 5


train batches: 100%|     | 326/326 [07:09<00:00,  1.32s/it, loss: 1.170]

Average epoch loss: 0.576

dev batches: 100%|     | 40/40 [00:15<00:00,  2.55it/s,  acc: 0.741, f1:
0.783]

Average epoch accuracy: 0.712
Average epoch f1: 0.763
Epoch: 3 of 5


train batches: 100%|     | 326/326 [07:10<00:00,  1.32s/it, loss: 0.063]

Average epoch loss: 0.299

dev batches: 100%|     | 40/40 [00:15<00:00,  2.54it/s,  acc: 0.739, f1:
0.781]

Average epoch accuracy: 0.739
Average epoch f1: 0.782
Epoch: 4 of 5


train batches: 100%|     | 326/326 [07:10<00:00,  1.32s/it, loss: 0.013]

Average epoch loss: 0.220

dev batches: 100%|     | 40/40 [00:15<00:00,  2.55it/s,  acc: 0.735, f1:
0.781]

Average epoch accuracy: 0.737
Average epoch f1: 0.781
Epoch: 5 of 5


train batches: 100%|     | 326/326 [07:10<00:00,  1.32s/it, loss: 0.018]

Average epoch loss: 0.136

dev batches: 100%|     | 40/40 [00:15<00:00,  2.55it/s,  acc: 0.735, f1:
0.781]

Average epoch accuracy: 0.736
Average epoch f1: 0.782
Done!
2023-05-07 15:01:52 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 5, run_best_acc: 0.738613486289978, run_best_f1:
0.7822245359420776

2023-05-07 15:01:52 model_05_cross_encoder_retrieval:INFO
```

```
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 15:01:52 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 15:01:54 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 15:01:54 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 344268.79it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 267795.53it/s]

generated dataset n=953
Epoch: 1 of 5


train batches:  61%|       | 198/326 [04:25<02:50,  1.34s/it, loss: 1.082]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
```

```
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [07:15<00:00,  1.33s/it, loss: 0.727]

Average epoch loss: 1.522

dev batches: 100%|     | 40/40 [00:15<00:00,  2.57it/s,  acc: 0.706, f1:
0.770]

Average epoch accuracy: 0.709
Average epoch f1: 0.776
Epoch: 2 of 5


train batches:  57%|       | 186/326 [04:07<03:06,  1.33s/it, loss: 0.732]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [07:12<00:00,  1.33s/it, loss: 0.057]

Average epoch loss: 0.523

dev batches: 100%|     | 40/40 [00:15<00:00,  2.57it/s,  acc: 0.722, f1:
0.777]

Average epoch accuracy: 0.718
Average epoch f1: 0.777
Epoch: 3 of 5


train batches:  27%|        | 87/326 [01:56<05:18,  1.33s/it, loss: 0.298]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [07:13<00:00,  1.33s/it, loss: 0.482]

Average epoch loss: 0.283

dev batches: 100%|     | 40/40 [00:15<00:00,  2.58it/s,  acc: 0.680, f1:
0.749]

Average epoch accuracy: 0.698
Average epoch f1: 0.761
Epoch: 4 of 5


train batches:  56%|       | 181/326 [04:00<03:14,  1.34s/it, loss: 0.057]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [07:13<00:00,  1.33s/it, loss: 0.006]

Average epoch loss: 0.176
```

```
dev batches: 100%|      | 40/40 [00:15<00:00,  2.56it/s,  acc: 0.659, f1:
0.738]

Average epoch accuracy: 0.670
Average epoch f1: 0.744
Epoch: 5 of 5


train batches:   5%|          | 17/326 [00:22<06:52,  1.34s/it, loss: 0.043]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [07:11<00:00,  1.32s/it, loss: 0.006]

Average epoch loss: 0.108

dev batches: 100%|      | 40/40 [00:15<00:00,  2.57it/s,  acc: 0.654, f1:
0.735]

Average epoch accuracy: 0.657
Average epoch f1: 0.737
Done!
2023-05-07 15:39:19 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 2, run_best_acc: 0.718364953994751, run_best_f1:
0.7765060663223267

2023-05-07 15:39:19 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 15:39:19 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).
```

```
2023-05-07 15:39:21 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 15:39:21 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 161719.53it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 334848.53it/s]

generated dataset n=953
Epoch: 1 of 5


train batches:   7%|         | 23/326 [00:31<06:39,  1.32s/it, loss: 2.083]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [07:11<00:00,  1.32s/it, loss: 0.343]

Average epoch loss: 1.475

dev batches: 100%|     | 40/40 [00:15<00:00,  2.56it/s,  acc: 0.785, f1:
0.817]

Average epoch accuracy: 0.785
Average epoch f1: 0.820
Epoch: 2 of 5


train batches:  58%|       | 190/326 [04:11<03:01,  1.33s/it, loss: 0.050]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [07:10<00:00,  1.32s/it, loss: 0.152]

Average epoch loss: 0.548

dev batches: 100%|     | 40/40 [00:15<00:00,  2.51it/s,  acc: 0.732, f1:
0.782]

Average epoch accuracy: 0.749
Average epoch f1: 0.793
```

```
Epoch: 3 of 5


train batches:  55%|       | 178/326 [03:55<03:15,  1.32s/it, loss: 0.986]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [07:09<00:00,  1.32s/it, loss: 0.535]

Average epoch loss: 0.250

dev batches: 100%|      | 40/40 [00:15<00:00,  2.57it/s,  acc: 0.678, f1:
0.749]

Average epoch accuracy: 0.703
Average epoch f1: 0.764
Epoch: 4 of 5


train batches:  99%|      | 324/326 [07:07<00:02,  1.32s/it, loss: 0.080]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [07:09<00:00,  1.32s/it, loss: 0.009]

Average epoch loss: 0.185

dev batches: 100%|      | 40/40 [00:15<00:00,  2.56it/s,  acc: 0.644, f1:
0.730]

Average epoch accuracy: 0.660
Average epoch f1: 0.739
Epoch: 5 of 5


train batches:  53%|       | 173/326 [03:48<03:22,  1.33s/it, loss: 0.118]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [07:10<00:00,  1.32s/it, loss: 0.016]

Average epoch loss: 0.158

dev batches: 100%|      | 40/40 [00:15<00:00,  2.54it/s,  acc: 0.640, f1:
0.727]

Average epoch accuracy: 0.642
Average epoch f1: 0.729
Done!
2023-05-07 16:16:31 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.7852210998535156, run_best_f1:
0.8198814392089844

2023-05-07 16:16:31 model_05_cross_encoder_retrieval:INFO
```

```
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 16:16:31 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 16:16:33 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 16:16:33 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 5, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 134991.62it/s]

generated dataset n=10260
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 245972.13it/s]

generated dataset n=1261
Epoch: 1 of 5


train batches: 100%|     | 428/428 [09:33<00:00,  1.34s/it, loss: 0.990]

Average epoch loss: 1.626
```

```
dev batches: 100%|     | 53/53 [00:20<00:00,  2.54it/s,  acc: 0.715, f1:
0.707]

Average epoch accuracy: 0.730
Average epoch f1: 0.725
Epoch: 2 of 5


train batches: 100%|     | 428/428 [09:41<00:00,  1.36s/it, loss: 0.052]

Average epoch loss: 0.581

dev batches: 100%|     | 53/53 [00:21<00:00,  2.45it/s,  acc: 0.742, f1:
0.718]

Average epoch accuracy: 0.735
Average epoch f1: 0.718
Epoch: 3 of 5


train batches: 100%|     | 428/428 [09:42<00:00,  1.36s/it, loss: 0.026]

Average epoch loss: 0.245

dev batches: 100%|     | 53/53 [00:21<00:00,  2.48it/s,  acc: 0.703, f1:
0.696]

Average epoch accuracy: 0.720
Average epoch f1: 0.706
Epoch: 4 of 5


train batches: 100%|     | 428/428 [09:42<00:00,  1.36s/it, loss: 0.018]

Average epoch loss: 0.203

dev batches: 100%|     | 53/53 [00:21<00:00,  2.47it/s,  acc: 0.677, f1:
0.679]

Average epoch accuracy: 0.689
Average epoch f1: 0.687
Epoch: 5 of 5


train batches: 100%|     | 428/428 [09:43<00:00,  1.36s/it, loss: 0.004]

Average epoch loss: 0.163

dev batches: 100%|     | 53/53 [00:21<00:00,  2.46it/s,  acc: 0.656, f1:
0.668]

Average epoch accuracy: 0.666
Average epoch f1: 0.673
Done!
2023-05-07 17:06:45 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.7351115942001343, run_best_f1:
```

```
0.724618136882782

2023-05-07 17:06:45 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 17:06:45 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 17:06:52 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 17:06:52 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 5, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|      | 1228/1228 [00:00<00:00, 148685.24it/s]

generated dataset n=10260
Torch device is 'mps'

claims: 100%|      | 154/154 [00:00<00:00, 198684.35it/s]

generated dataset n=1261
Epoch: 1 of 5
```

```
train batches: 100%|     | 428/428 [09:32<00:00,  1.34s/it, loss: 0.759]

Average epoch loss: 1.454

dev batches: 100%|     | 53/53 [00:20<00:00,  2.52it/s,  acc: 0.695, f1:
0.684]

Average epoch accuracy: 0.696
Average epoch f1: 0.689
Epoch: 2 of 5


train batches: 100%|     | 428/428 [09:33<00:00,  1.34s/it, loss: 0.075]

Average epoch loss: 0.547

dev batches: 100%|     | 53/53 [00:20<00:00,  2.53it/s,  acc: 0.743, f1:
0.721]

Average epoch accuracy: 0.721
Average epoch f1: 0.706
Epoch: 3 of 5


train batches: 100%|     | 428/428 [09:34<00:00,  1.34s/it, loss: 0.017]

Average epoch loss: 0.308

dev batches: 100%|     | 53/53 [00:20<00:00,  2.54it/s,  acc: 0.706, f1:
0.699]

Average epoch accuracy: 0.723
Average epoch f1: 0.710
Epoch: 4 of 5


train batches: 100%|     | 428/428 [09:35<00:00,  1.34s/it, loss: 0.010]

Average epoch loss: 0.220

dev batches: 100%|     | 53/53 [00:21<00:00,  2.51it/s,  acc: 0.670, f1:
0.679]

Average epoch accuracy: 0.686
Average epoch f1: 0.688
Epoch: 5 of 5


train batches: 100%|     | 428/428 [09:36<00:00,  1.35s/it, loss: 0.014]

Average epoch loss: 0.158

dev batches: 100%|     | 53/53 [00:21<00:00,  2.50it/s,  acc: 0.661, f1:
0.673]

Average epoch accuracy: 0.665
Average epoch f1: 0.676
```

```
Done!
2023-05-07 17:56:32 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 3, run_best_acc: 0.7225133776664734, run_best_f1:
0.71006840467453

2023-05-07 17:56:32 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 17:56:32 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 17:56:33 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 17:56:33 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 5, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 147615.65it/s]

generated dataset n=10260
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 264073.11it/s]
```

```
generated dataset n=1261
Epoch: 1 of 5


train batches:  44%|       | 188/428 [04:08<05:14,  1.31s/it, loss: 1.511]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  46%|       | 195/428 [04:17<05:10,  1.33s/it, loss: 1.335]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  61%|       | 261/428 [05:44<03:40,  1.32s/it, loss: 1.099]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  85%|      | 362/428 [07:56<01:25,  1.30s/it, loss: 1.061]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  86%|      | 370/428 [08:06<01:15,  1.30s/it, loss: 1.090]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  92%|     | 393/428 [08:36<00:45,  1.31s/it, loss: 0.821]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  94%|     | 401/428 [08:47<00:35,  1.30s/it, loss: 0.734]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:21<00:00,  1.31s/it, loss: 0.643]

Average epoch loss: 1.522

dev batches: 100%|     | 53/53 [00:20<00:00,  2.62it/s,  acc: 0.642, f1:
0.669]

Average epoch accuracy: 0.658
Average epoch f1: 0.687
Epoch: 2 of 5


train batches:  18%|         | 77/428 [01:39<07:34,  1.29s/it, loss: 0.894]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  25%|         | 105/428 [02:15<06:57,  1.29s/it, loss: 0.203]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
```

```
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  27%|        | 114/428 [02:27<06:47,  1.30s/it, loss: 0.920]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  32%|        | 137/428 [02:57<06:15,  1.29s/it, loss: 0.519]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  59%|       | 253/428 [05:27<03:46,  1.29s/it, loss: 0.641]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  73%|      | 313/428 [06:45<02:28,  1.30s/it, loss: 0.141]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  92%|     | 395/428 [08:31<00:42,  1.28s/it, loss: 0.263]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:12<00:00,  1.29s/it, loss: 0.177]

Average epoch loss: 0.494

dev batches: 100%|     | 53/53 [00:19<00:00,  2.65it/s,  acc: 0.670, f1:
0.677]

Average epoch accuracy: 0.659
Average epoch f1: 0.675
Epoch: 3 of 5


train batches:  14%|         | 60/428 [01:16<07:50,  1.28s/it, loss: 1.208]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  50%|       | 215/428 [04:34<04:32,  1.28s/it, loss: 0.902]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  60%|       | 258/428 [05:29<03:40,  1.30s/it, loss: 0.245]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
Be aware, overflowing tokens are not returned for the setting you have chosen,
i.e. sequence pairs with the 'longest_first' truncation strategy. So the
returned list will always be empty even if some tokens have been removed.
train batches:  76%|      | 326/428 [06:58<02:14,  1.31s/it, loss: 0.383]Be
```

```
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  85%|      | 365/428 [07:49<01:23,  1.32s/it, loss: 0.239]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  89%|      | 380/428 [08:09<01:03,  1.32s/it, loss: 0.237]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:11<00:00,  1.29s/it, loss: 0.147]

Average epoch loss: 0.296

dev batches: 100%|     | 53/53 [00:20<00:00,  2.61it/s,  acc: 0.654, f1:
0.671]

Average epoch accuracy: 0.660
Average epoch f1: 0.673
Epoch: 4 of 5


train batches:   5%|         | 21/428 [00:27<08:49,  1.30s/it, loss: 0.202]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  20%|         | 87/428 [01:53<07:25,  1.31s/it, loss: 0.226]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  28%|        | 119/428 [02:35<06:42,  1.30s/it, loss: 0.015]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  54%|       | 229/428 [04:58<04:17,  1.29s/it, loss: 0.172]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  64%|       | 276/428 [05:59<03:16,  1.30s/it, loss: 0.101]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  68%|       | 292/428 [06:20<02:56,  1.30s/it, loss: 0.488]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  75%|      | 320/428 [06:56<02:20,  1.30s/it, loss: 0.016]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
```

```
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:16<00:00,  1.30s/it, loss: 0.351]

Average epoch loss: 0.200

dev batches: 100%|     | 53/53 [00:20<00:00,  2.62it/s,  acc: 0.638, f1:
0.659]

Average epoch accuracy: 0.646
Average epoch f1: 0.665
Epoch: 5 of 5


train batches:   2%|         | 7/428 [00:09<09:09,  1.31s/it, loss: 0.068]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:   7%|         | 29/428 [00:37<08:37,  1.30s/it, loss: 1.912]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  17%|        | 72/428 [01:33<07:43,  1.30s/it, loss: 0.025]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  41%|        | 174/428 [03:47<05:29,  1.30s/it, loss: 0.042]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  72%|      | 309/428 [06:42<02:34,  1.30s/it, loss: 0.017]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  88%|      | 375/428 [08:08<01:09,  1.30s/it, loss: 0.044]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  89%|      | 379/428 [08:13<01:04,  1.32s/it, loss: 0.002]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:17<00:00,  1.30s/it, loss: 0.004]

Average epoch loss: 0.131

dev batches: 100%|     | 53/53 [00:20<00:00,  2.61it/s,  acc: 0.611, f1:
0.644]

Average epoch accuracy: 0.623
Average epoch f1: 0.651
Done!
```

```
2023-05-07 18:44:37 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.6596076488494873, run_best_f1:
0.687384843826294

2023-05-07 18:44:37 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 18:44:37 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 18:44:39 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 18:44:39 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 5, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 10706.94it/s]

generated dataset n=10260
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 259302.62it/s]

generated dataset n=1261
```

```
Epoch: 1 of 5


train batches:   8%|         | 34/428 [00:45<08:32,  1.30s/it, loss: 2.579]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  11%|         | 45/428 [00:59<08:20,  1.31s/it, loss: 1.606]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  28%|        | 118/428 [02:35<06:45,  1.31s/it, loss: 0.973]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  36%|        | 152/428 [03:19<06:03,  1.32s/it, loss: 1.614]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  68%|       | 293/428 [06:23<02:56,  1.31s/it, loss: 1.002]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  75%|      | 323/428 [07:03<02:16,  1.30s/it, loss: 0.215]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  79%|      | 340/428 [07:25<01:54,  1.30s/it, loss: 0.685]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:19<00:00,  1.31s/it, loss: 2.546]

Average epoch loss: 1.436

dev batches: 100%|     | 53/53 [00:20<00:00,  2.61it/s,  acc: 0.746, f1:
0.733]

Average epoch accuracy: 0.738
Average epoch f1: 0.731
Epoch: 2 of 5


train batches:   5%|         | 22/428 [00:28<08:50,  1.31s/it, loss: 0.092]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:   6%|         | 26/428 [00:34<08:43,  1.30s/it, loss: 0.395]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
```

```
list will always be empty even if some tokens have been removed.
train batches:  39%|        | 166/428 [03:37<05:44,  1.32s/it, loss: 0.074]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  54%|       | 231/428 [05:02<04:16,  1.30s/it, loss: 1.091]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  60%|       | 258/428 [05:37<03:40,  1.29s/it, loss: 0.718]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  86%|      | 367/428 [07:59<01:19,  1.31s/it, loss: 0.294]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  89%|      | 381/428 [08:18<01:01,  1.31s/it, loss: 0.039]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:18<00:00,  1.30s/it, loss: 0.109]

Average epoch loss: 0.529

dev batches: 100%|     | 53/53 [00:20<00:00,  2.65it/s,  acc: 0.693, f1:
0.695]

Average epoch accuracy: 0.715
Average epoch f1: 0.710
Epoch: 3 of 5


train batches:   1%|          | 4/428 [00:05<09:06,  1.29s/it, loss: 0.067]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  12%|         | 51/428 [01:05<08:01,  1.28s/it, loss: 0.028]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  15%|         | 66/428 [01:24<07:42,  1.28s/it, loss: 0.982]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  18%|         | 77/428 [01:39<07:33,  1.29s/it, loss: 0.398]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  81%|      | 346/428 [07:26<01:44,  1.28s/it, loss: 0.498]Be
```

```
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  96%|      | 409/428 [08:47<00:24,  1.29s/it, loss: 0.011]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  99%|      | 424/428 [09:06<00:05,  1.29s/it, loss: 0.286]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 428/428 [09:11<00:00,  1.29s/it, loss: 0.216]

Average epoch loss: 0.267

dev batches: 100%|      | 53/53 [00:19<00:00,  2.65it/s,  acc: 0.655, f1:
0.670]

Average epoch accuracy: 0.672
Average epoch f1: 0.681
Epoch: 4 of 5


train batches:  10%|          | 43/428 [00:55<08:11,  1.28s/it, loss: 0.109]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  25%|         | 106/428 [02:16<06:51,  1.28s/it, loss: 1.065]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
Be aware, overflowing tokens are not returned for the setting you have chosen,
i.e. sequence pairs with the 'longest_first' truncation strategy. So the
returned list will always be empty even if some tokens have been removed.
train batches:  47%|        | 203/428 [04:21<04:49,  1.28s/it, loss: 0.078]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  68%|       | 293/428 [06:16<02:52,  1.28s/it, loss: 0.090]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  74%|      | 317/428 [06:47<02:23,  1.29s/it, loss: 0.011]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  90%|      | 384/428 [08:13<00:56,  1.29s/it, loss: 0.095]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
```

```
train batches: 100%|     | 428/428 [09:09<00:00,  1.28s/it, loss: 0.020]

Average epoch loss: 0.177

dev batches: 100%|     | 53/53 [00:19<00:00,  2.65it/s,  acc: 0.648, f1:
0.667]

Average epoch accuracy: 0.651
Average epoch f1: 0.669
Epoch: 5 of 5


train batches:  39%|        | 169/428 [03:37<05:34,  1.29s/it, loss: 0.009]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  48%|        | 206/428 [04:25<04:44,  1.28s/it, loss: 0.071]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  52%|       | 221/428 [04:44<04:27,  1.29s/it, loss: 0.128]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  57%|       | 242/428 [05:11<03:59,  1.29s/it, loss: 0.036]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  57%|       | 245/428 [05:15<03:57,  1.30s/it, loss: 0.020]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  84%|      | 359/428 [07:42<01:28,  1.29s/it, loss: 0.052]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  95%|     | 407/428 [08:44<00:27,  1.29s/it, loss: 0.087]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 428/428 [09:10<00:00,  1.29s/it, loss: 0.003]

Average epoch loss: 0.172

dev batches: 100%|     | 53/53 [00:20<00:00,  2.65it/s,  acc: 0.628, f1:
0.656]

Average epoch accuracy: 0.637
Average epoch f1: 0.661
Done!
2023-05-07 19:32:30 model_05_cross_encoder_retrieval:INFO
```

```
run_best_epoch: 1, run_best_acc: 0.7382638454437256, run_best_f1:
0.7312518954277039

2023-05-07 19:32:30 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 19:32:30 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 19:32:36 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 19:32:36 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 10, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 120237.30it/s]

generated dataset n=16395
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 180123.48it/s]

generated dataset n=2031
Epoch: 1 of 5
```

```
train batches: 100%|     | 684/684 [14:42<00:00,  1.29s/it, loss: 0.731]

Average epoch loss: 1.333

dev batches: 100%|     | 85/85 [00:32<00:00,  2.61it/s,  acc: 0.650, f1:
0.550]

Average epoch accuracy: 0.655
Average epoch f1: 0.567
Epoch: 2 of 5


train batches: 100%|     | 684/684 [14:39<00:00,  1.29s/it, loss: 0.704]

Average epoch loss: 0.500

dev batches: 100%|     | 85/85 [00:32<00:00,  2.61it/s,  acc: 0.698, f1:
0.580]

Average epoch accuracy: 0.679
Average epoch f1: 0.570
Epoch: 3 of 5


train batches: 100%|     | 684/684 [14:38<00:00,  1.28s/it, loss: 0.032]

Average epoch loss: 0.282

dev batches: 100%|     | 85/85 [00:32<00:00,  2.62it/s,  acc: 0.670, f1:
0.561]

Average epoch accuracy: 0.681
Average epoch f1: 0.570
Epoch: 4 of 5


train batches: 100%|     | 684/684 [14:37<00:00,  1.28s/it, loss: 0.005]

Average epoch loss: 0.176

dev batches: 100%|     | 85/85 [00:32<00:00,  2.61it/s,  acc: 0.647, f1:
0.540]

Average epoch accuracy: 0.657
Average epoch f1: 0.551
Epoch: 5 of 5


train batches: 100%|     | 684/684 [14:39<00:00,  1.29s/it, loss: 0.146]

Average epoch loss: 0.169

dev batches: 100%|     | 85/85 [00:32<00:00,  2.62it/s,  acc: 0.658, f1:
0.547]
```

```
Average epoch accuracy: 0.652
Average epoch f1: 0.544
Done!
2023-05-07 20:48:37 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 2, run_best_acc: 0.6814714074134827, run_best_f1:
0.5700081586837769

2023-05-07 20:48:37 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 20:48:37 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 20:48:39 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 20:48:39 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 10, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 121596.99it/s]

generated dataset n=16395
Torch device is 'mps'
```

```
claims: 100%|      | 154/154 [00:00<00:00, 172614.33it/s]

generated dataset n=2031
Epoch: 1 of 5


train batches: 100%|      | 684/684 [14:48<00:00,  1.30s/it, loss: 0.185]

Average epoch loss: 1.329

dev batches: 100%|      | 85/85 [00:33<00:00,  2.57it/s,  acc: 0.721, f1:
0.599]

Average epoch accuracy: 0.730
Average epoch f1: 0.619
Epoch: 2 of 5


train batches: 100%|      | 684/684 [14:55<00:00,  1.31s/it, loss: 0.161]

Average epoch loss: 0.548

dev batches: 100%|      | 85/85 [00:32<00:00,  2.62it/s,  acc: 0.713, f1:
0.592]

Average epoch accuracy: 0.717
Average epoch f1: 0.598
Epoch: 3 of 5


train batches: 100%|      | 684/684 [14:37<00:00,  1.28s/it, loss: 0.036]

Average epoch loss: 0.303

dev batches: 100%|      | 85/85 [00:32<00:00,  2.62it/s,  acc: 0.692, f1:
0.581]

Average epoch accuracy: 0.700
Average epoch f1: 0.587
Epoch: 4 of 5


train batches: 100%|      | 684/684 [14:37<00:00,  1.28s/it, loss: 0.014]

Average epoch loss: 0.191

dev batches: 100%|      | 85/85 [00:32<00:00,  2.62it/s,  acc: 0.696, f1:
0.579]

Average epoch accuracy: 0.695
Average epoch f1: 0.581
Epoch: 5 of 5


train batches: 100%|      | 684/684 [14:36<00:00,  1.28s/it, loss: 0.002]

Average epoch loss: 0.155
```

```
dev batches: 100%|      | 85/85 [00:32<00:00,  2.62it/s,  acc: 0.684, f1:
0.569]

Average epoch accuracy: 0.690
Average epoch f1: 0.575
Done!
2023-05-07 22:04:59 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.7300872206687927, run_best_f1:
0.619188666343689

2023-05-07 22:04:59 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 22:04:59 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).

2023-05-07 22:05:01 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 22:05:01 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 10, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|      | 1228/1228 [00:00<00:00, 120226.08it/s]
```

```
generated dataset n=16395
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 173030.49it/s]

generated dataset n=2031
Epoch: 1 of 5


train batches:   5%|         | 33/684 [00:43<14:00,  1.29s/it, loss: 3.001]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  11%|         | 73/684 [01:35<13:07,  1.29s/it, loss: 2.748]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  13%|         | 88/684 [01:54<12:46,  1.29s/it, loss: 1.871]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  17%|         | 115/684 [02:29<12:11,  1.29s/it, loss: 1.933]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  26%|        | 179/684 [03:52<10:52,  1.29s/it, loss: 1.001]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  35%|        | 240/684 [05:10<09:31,  1.29s/it, loss: 0.574]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  60%|       | 408/684 [08:47<05:57,  1.29s/it, loss: 0.971]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  71%|       | 486/684 [10:28<04:16,  1.29s/it, loss: 1.539]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  75%|      | 514/684 [11:04<03:39,  1.29s/it, loss: 1.069]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
Be aware, overflowing tokens are not returned for the setting you have chosen,
i.e. sequence pairs with the 'longest_first' truncation strategy. So the
returned list will always be empty even if some tokens have been removed.
train batches:  85%|      | 582/684 [12:32<02:12,  1.29s/it, loss: 0.596]Be
```

```
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  86%|      | 590/684 [12:43<02:01,  1.30s/it, loss: 0.668]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  88%|      | 603/684 [12:59<01:44,  1.30s/it, loss: 1.892]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  96%|      | 654/684 [14:05<00:38,  1.29s/it, loss: 0.556]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  96%|      | 660/684 [14:13<00:31,  1.30s/it, loss: 1.713]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 684/684 [14:43<00:00,  1.29s/it, loss: 0.801]

Average epoch loss: 1.299

dev batches: 100%|      | 85/85 [00:32<00:00,  2.63it/s,  acc: 0.670, f1:
0.564]

Average epoch accuracy: 0.673
Average epoch f1: 0.579
Epoch: 2 of 5


train batches:  13%|         | 91/684 [01:57<12:42,  1.29s/it, loss: 0.655]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  21%|         | 141/684 [03:01<11:40,  1.29s/it, loss: 1.694]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  21%|         | 146/684 [03:08<11:34,  1.29s/it, loss: 0.172]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  33%|        | 223/684 [04:47<09:55,  1.29s/it, loss: 0.115]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  40%|        | 274/684 [05:53<08:48,  1.29s/it, loss: 0.083]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
```

```
list will always be empty even if some tokens have been removed.
train batches:  44%|        | 304/684 [06:32<08:09,  1.29s/it, loss: 0.570]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  58%|       | 399/684 [08:34<06:07,  1.29s/it, loss: 0.063]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  61%|       | 416/684 [08:56<05:45,  1.29s/it, loss: 0.265]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  67%|       | 460/684 [09:53<04:47,  1.29s/it, loss: 0.516]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  75%|      | 514/684 [11:02<03:38,  1.29s/it, loss: 0.902]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  84%|      | 573/684 [12:18<02:22,  1.28s/it, loss: 1.204]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  85%|      | 583/684 [12:31<02:10,  1.29s/it, loss: 0.668]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  88%|      | 605/684 [13:00<01:41,  1.29s/it, loss: 0.258]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  94%|     | 641/684 [13:46<00:55,  1.29s/it, loss: 0.077]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  96%|     | 656/684 [14:05<00:36,  1.29s/it, loss: 0.331]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:40<00:00,  1.29s/it, loss: 0.091]

Average epoch loss: 0.505

dev batches: 100%|     | 85/85 [00:32<00:00,  2.63it/s,  acc: 0.712, f1:
0.590]

Average epoch accuracy: 0.694
```

```
Average epoch f1: 0.580
Epoch: 3 of 5


train batches:   1%|          | 5/684 [00:06<14:36,  1.29s/it, loss: 0.040]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:   9%|         | 60/684 [01:17<13:25,  1.29s/it, loss: 0.888]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  23%|        | 157/684 [03:22<11:19,  1.29s/it, loss: 0.031]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  31%|        | 212/684 [04:33<10:08,  1.29s/it, loss: 0.134]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  32%|        | 219/684 [04:42<10:00,  1.29s/it, loss: 0.026]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  37%|        | 253/684 [05:26<09:16,  1.29s/it, loss: 0.039]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  55%|       | 378/684 [08:07<06:35,  1.29s/it, loss: 0.300]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  69%|       | 470/684 [10:06<04:35,  1.29s/it, loss: 0.081]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  70%|       | 476/684 [10:13<04:28,  1.29s/it, loss: 0.136]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  76%|      | 518/684 [11:08<03:34,  1.29s/it, loss: 0.047]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  82%|      | 560/684 [12:02<02:40,  1.29s/it, loss: 0.048]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
```

```
train batches:  87%|      | 598/684 [12:51<01:50,  1.29s/it, loss: 1.377]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  93%|      | 638/684 [13:42<00:59,  1.29s/it, loss: 0.601]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  97%|      | 664/684 [14:16<00:25,  1.29s/it, loss: 0.004]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 682/684 [14:39<00:02,  1.29s/it, loss: 0.758]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 684/684 [14:41<00:00,  1.29s/it, loss: 0.008]

Average epoch loss: 0.268

dev batches: 100%|      | 85/85 [00:32<00:00,  2.63it/s,  acc: 0.676, f1:
0.560]

Average epoch accuracy: 0.692
Average epoch f1: 0.574
Epoch: 4 of 5


train batches:  19%|         | 131/684 [02:48<11:49,  1.28s/it, loss: 0.010]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  20%|         | 140/684 [03:00<11:40,  1.29s/it, loss: 0.214]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  22%|         | 153/684 [03:16<11:23,  1.29s/it, loss: 0.015]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  26%|         | 175/684 [03:45<10:57,  1.29s/it, loss: 0.020]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  29%|         | 195/684 [04:10<10:29,  1.29s/it, loss: 0.005]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  30%|         | 207/684 [04:26<10:15,  1.29s/it, loss: 0.002]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
```

```
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  38%|        | 261/684 [05:35<09:02,  1.28s/it, loss: 0.072]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  43%|        | 297/684 [06:22<08:17,  1.29s/it, loss: 0.237]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  54%|       | 367/684 [07:52<06:46,  1.28s/it, loss: 0.062]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  66%|       | 451/684 [09:40<04:59,  1.28s/it, loss: 0.004]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
Be aware, overflowing tokens are not returned for the setting you have chosen,
i.e. sequence pairs with the 'longest_first' truncation strategy. So the
returned list will always be empty even if some tokens have been removed.
train batches:  71%|       | 483/684 [10:21<04:17,  1.28s/it, loss: 0.038]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  76%|      | 520/684 [11:08<03:31,  1.29s/it, loss: 0.329]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  78%|      | 531/684 [11:22<03:17,  1.29s/it, loss: 0.004]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  93%|     | 634/684 [13:35<01:04,  1.29s/it, loss: 0.086]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:38<00:00,  1.28s/it, loss: 0.014]

Average epoch loss: 0.190

dev batches: 100%|     | 85/85 [00:32<00:00,  2.64it/s,  acc: 0.687, f1:
0.562]

Average epoch accuracy: 0.682
Average epoch f1: 0.562
Epoch: 5 of 5


train batches:   6%|         | 40/684 [00:51<13:49,  1.29s/it, loss: 0.003]Be
```

```
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  14%|         | 94/684 [02:01<12:40,  1.29s/it, loss: 0.011]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  20%|         | 135/684 [02:54<11:47,  1.29s/it, loss: 0.049]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  24%|         | 161/684 [03:27<11:12,  1.29s/it, loss: 0.043]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  24%|         | 164/684 [03:31<11:09,  1.29s/it, loss: 0.023]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  38%|        | 261/684 [05:36<09:04,  1.29s/it, loss: 0.002]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  61%|       | 418/684 [08:59<05:43,  1.29s/it, loss: 0.009]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  68%|       | 464/684 [09:58<04:42,  1.29s/it, loss: 0.098]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  68%|       | 467/684 [10:02<04:39,  1.29s/it, loss: 0.020]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  70%|       | 482/684 [10:21<04:20,  1.29s/it, loss: 0.009]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  71%|      | 489/684 [10:30<04:11,  1.29s/it, loss: 0.148]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  83%|      | 565/684 [12:08<02:33,  1.29s/it, loss: 0.035]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  88%|      | 600/684 [12:53<01:48,  1.29s/it, loss: 0.009]Be
```

```
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  90%|      | 613/684 [13:10<01:31,  1.29s/it, loss: 0.476]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 681/684 [14:38<00:03,  1.29s/it, loss: 0.054]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:40<00:00,  1.29s/it, loss: 0.001]

Average epoch loss: 0.156

dev batches: 100%|     | 85/85 [00:32<00:00,  2.63it/s,  acc: 0.652, f1:
0.537]

Average epoch accuracy: 0.667
Average epoch f1: 0.548
Done!
2023-05-07 23:21:09 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 2, run_best_acc: 0.6937678456306458, run_best_f1:
0.5799320936203003

2023-05-07 23:21:09 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-07 23:21:09 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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).
```

```
2023-05-07 23:21:11 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-07 23:21:11 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 10, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 125306.67it/s]

generated dataset n=16395
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 182206.72it/s]

generated dataset n=2031
Epoch: 1 of 5


train batches:  22%|        | 152/684 [03:16<11:20,  1.28s/it, loss: 1.124]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  23%|        | 154/684 [03:18<11:20,  1.28s/it, loss: 1.279]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  29%|        | 200/684 [04:17<10:23,  1.29s/it, loss: 0.976]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  30%|        | 203/684 [04:21<10:20,  1.29s/it, loss: 1.940]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  34%|        | 235/684 [05:03<09:38,  1.29s/it, loss: 1.250]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  35%|        | 240/684 [05:09<09:31,  1.29s/it, loss: 1.027]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  45%|        | 311/684 [06:41<08:01,  1.29s/it, loss: 0.936]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
```

```
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  52%|       | 358/684 [07:41<07:00,  1.29s/it, loss: 2.111]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  58%|       | 399/684 [08:34<06:07,  1.29s/it, loss: 1.254]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  65%|       | 444/684 [09:32<05:08,  1.28s/it, loss: 0.146]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  79%|      | 537/684 [11:31<03:09,  1.29s/it, loss: 0.481]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  88%|      | 601/684 [12:54<01:46,  1.29s/it, loss: 1.429]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  88%|      | 603/684 [12:56<01:44,  1.29s/it, loss: 1.542]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  94%|     | 644/684 [13:49<00:51,  1.28s/it, loss: 0.246]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  95%|     | 650/684 [13:57<00:43,  1.29s/it, loss: 1.280]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:39<00:00,  1.29s/it, loss: 0.058]

Average epoch loss: 1.279

dev batches: 100%|     | 85/85 [00:32<00:00,  2.63it/s,  acc: 0.542, f1:
0.486]

Average epoch accuracy: 0.550
Average epoch f1: 0.505
Epoch: 2 of 5


train batches:   8%|         | 57/684 [01:13<13:24,  1.28s/it, loss: 0.890]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
```

```
train batches:  13%|         | 88/684 [01:53<12:45,  1.28s/it, loss: 0.126]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  14%|         | 97/684 [02:05<12:32,  1.28s/it, loss: 0.977]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  16%|         | 108/684 [02:19<12:22,  1.29s/it, loss: 0.390]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  16%|         | 111/684 [02:23<12:23,  1.30s/it, loss: 0.210]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  36%|        | 244/684 [05:14<09:28,  1.29s/it, loss: 0.509]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  46%|        | 315/684 [06:46<07:56,  1.29s/it, loss: 0.205]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  51%|       | 351/684 [07:32<07:08,  1.29s/it, loss: 0.085]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  61%|       | 420/684 [09:01<05:38,  1.28s/it, loss: 0.299]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  79%|      | 541/684 [11:36<03:03,  1.28s/it, loss: 0.489]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  83%|      | 568/684 [12:10<02:28,  1.28s/it, loss: 1.743]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  90%|      | 617/684 [13:13<01:25,  1.28s/it, loss: 0.079]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  94%|     | 645/684 [13:49<00:49,  1.28s/it, loss: 0.116]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
```

```
train batches:  96%|     | 655/684 [14:02<00:37,  1.28s/it, loss: 1.579]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  97%|     | 662/684 [14:11<00:28,  1.29s/it, loss: 1.411]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:38<00:00,  1.28s/it, loss: 1.148]

Average epoch loss: 0.479

dev batches: 100%|     | 85/85 [00:32<00:00,  2.64it/s,  acc: 0.552, f1:
0.483]

Average epoch accuracy: 0.550
Average epoch f1: 0.488
Epoch: 3 of 5


train batches:   2%|         | 17/684 [00:21<14:16,  1.28s/it, loss: 0.032]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:   7%|         | 50/684 [01:04<13:30,  1.28s/it, loss: 0.194]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:   8%|         | 58/684 [01:14<13:20,  1.28s/it, loss: 0.135]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  26%|        | 180/684 [03:50<10:44,  1.28s/it, loss: 0.514]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  27%|        | 187/684 [03:59<10:35,  1.28s/it, loss: 1.553]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  28%|        | 191/684 [04:04<10:31,  1.28s/it, loss: 0.496]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  30%|        | 205/684 [04:22<10:13,  1.28s/it, loss: 0.054]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  35%|        | 242/684 [05:10<09:26,  1.28s/it, loss: 0.395]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
```

```
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  36%|        | 243/684 [05:11<09:25,  1.28s/it, loss: 0.035]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  47%|        | 322/684 [06:52<07:43,  1.28s/it, loss: 0.013]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  49%|        | 336/684 [07:10<07:26,  1.28s/it, loss: 0.668]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  64%|       | 440/684 [09:23<05:13,  1.28s/it, loss: 1.306]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  75%|      | 516/684 [11:01<03:35,  1.28s/it, loss: 0.056]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  78%|      | 533/684 [11:23<03:13,  1.28s/it, loss: 0.075]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  85%|      | 582/684 [12:25<02:10,  1.28s/it, loss: 0.057]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:35<00:00,  1.28s/it, loss: 0.019]

Average epoch loss: 0.285

dev batches: 100%|     | 85/85 [00:32<00:00,  2.64it/s,  acc: 0.575, f1:
0.492]

Average epoch accuracy: 0.564
Average epoch f1: 0.488
Epoch: 4 of 5


train batches:  13%|         | 88/684 [01:52<12:44,  1.28s/it, loss: 0.023]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  17%|         | 113/684 [02:24<12:10,  1.28s/it, loss: 0.025]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
```

```
train batches:  20%|         | 135/684 [02:53<11:44,  1.28s/it, loss: 0.606]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  23%|         | 155/684 [03:18<11:17,  1.28s/it, loss: 0.166]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  23%|         | 159/684 [03:23<11:11,  1.28s/it, loss: 0.031]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  29%|         | 199/684 [04:15<10:22,  1.28s/it, loss: 0.390]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  32%|         | 217/684 [04:38<09:58,  1.28s/it, loss: 0.127]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  33%|         | 227/684 [04:51<09:45,  1.28s/it, loss: 0.128]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  46%|         | 315/684 [06:43<07:52,  1.28s/it, loss: 0.988]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  52%|        | 355/684 [07:35<07:00,  1.28s/it, loss: 0.038]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  52%|        | 359/684 [07:40<06:56,  1.28s/it, loss: 0.003]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  68%|       | 465/684 [09:55<04:39,  1.28s/it, loss: 0.046]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  76%|       | 521/684 [11:07<03:28,  1.28s/it, loss: 0.464]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  77%|       | 528/684 [11:16<03:19,  1.28s/it, loss: 0.233]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
```

```
train batches:  82%|      | 558/684 [11:54<02:41,  1.28s/it, loss: 0.011]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:35<00:00,  1.28s/it, loss: 0.063]

Average epoch loss: 0.208

dev batches: 100%|     | 85/85 [00:32<00:00,  2.64it/s,  acc: 0.570, f1:
0.486]

Average epoch accuracy: 0.573
Average epoch f1: 0.490
Epoch: 5 of 5


train batches:   4%|         | 28/684 [00:35<14:02,  1.28s/it, loss: 0.177]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:   8%|         | 57/684 [01:13<13:22,  1.28s/it, loss: 0.154]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  10%|         | 70/684 [01:29<13:07,  1.28s/it, loss: 0.010]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  15%|         | 103/684 [02:12<12:24,  1.28s/it, loss: 0.007]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  23%|        | 158/684 [03:22<11:12,  1.28s/it, loss: 0.030]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  34%|        | 230/684 [04:54<09:42,  1.28s/it, loss: 0.003]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  55%|       | 379/684 [08:05<06:30,  1.28s/it, loss: 0.004]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  64%|       | 436/684 [09:18<05:17,  1.28s/it, loss: 0.064]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  65%|       | 445/684 [09:30<05:05,  1.28s/it, loss: 0.066]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
```

```
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  70%|       | 480/684 [10:14<04:21,  1.28s/it, loss: 0.439]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  72%|      | 491/684 [10:28<04:06,  1.28s/it, loss: 0.439]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  75%|      | 515/684 [10:59<03:36,  1.28s/it, loss: 0.007]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  88%|      | 603/684 [12:52<01:43,  1.28s/it, loss: 0.004]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  90%|      | 614/684 [13:06<01:29,  1.28s/it, loss: 0.296]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches:  96%|     | 658/684 [14:02<00:33,  1.28s/it, loss: 0.027]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 684/684 [14:35<00:00,  1.28s/it, loss: 0.001]

Average epoch loss: 0.147

dev batches: 100%|     | 85/85 [00:32<00:00,  2.64it/s,  acc: 0.572, f1:
0.487]

Average epoch accuracy: 0.572
Average epoch f1: 0.488
Done!
2023-05-08 00:36:58 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.5728825926780701, run_best_f1:
0.5045245289802551

2023-05-08 00:36:58 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-08 00:36:58 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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 00:37:06 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-08 00:37:06 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 10, 'n_neg_samples': 3, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 270842.16it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 266800.01it/s]

generated dataset n=953
Epoch: 1 of 10


train batches: 100%|     | 326/326 [06:58<00:00,  1.29s/it, loss: 0.838]

Average epoch loss: 1.551

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.721, f1:
0.764]

Average epoch accuracy: 0.718
Average epoch f1: 0.766
Epoch: 2 of 10
```

```
train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.281]

Average epoch loss: 0.585

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.700, f1:
0.756]

Average epoch accuracy: 0.708
Average epoch f1: 0.760
Epoch: 3 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.107]

Average epoch loss: 0.296

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.687, f1:
0.749]

Average epoch accuracy: 0.692
Average epoch f1: 0.752
Epoch: 4 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.052]

Average epoch loss: 0.194

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.652, f1:
0.729]

Average epoch accuracy: 0.668
Average epoch f1: 0.738
Epoch: 5 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.002]

Average epoch loss: 0.148

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.633, f1:
0.718]

Average epoch accuracy: 0.641
Average epoch f1: 0.723
Epoch: 6 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.002]

Average epoch loss: 0.127

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.633, f1:
0.720]

Average epoch accuracy: 0.633
Average epoch f1: 0.720
```

```
Epoch: 7 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.033]

Average epoch loss: 0.119

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.621, f1:
0.715]

Average epoch accuracy: 0.628
Average epoch f1: 0.718
Epoch: 8 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 3.003]

Average epoch loss: 0.132

dev batches: 100%|     | 40/40 [00:15<00:00,  2.62it/s,  acc: 0.618, f1:
0.714]

Average epoch accuracy: 0.619
Average epoch f1: 0.714
Epoch: 9 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.004]

Average epoch loss: 0.111

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.614, f1:
0.712]

Average epoch accuracy: 0.616
Average epoch f1: 0.713
Epoch: 10 of 10


train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.105]

Average epoch loss: 0.091

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.617, f1:
0.713]

Average epoch accuracy: 0.616
Average epoch f1: 0.713
Done!
2023-05-08 01:49:16 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.7180168032646179, run_best_f1:
0.7657316327095032

2023-05-08 01:49:16 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063
```

```
2023-05-08 01:49:16 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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 01:49:18 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-08 01:49:18 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 10, 'n_neg_samples': 3, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|      | 1228/1228 [00:00<00:00, 179707.80it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|      | 154/154 [00:00<00:00, 348958.84it/s]

generated dataset n=953
Epoch: 1 of 10


train batches: 100%|      | 326/326 [07:00<00:00,  1.29s/it, loss: 0.130]

Average epoch loss: 1.676
```

```
dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.699, f1:
0.767]

Average epoch accuracy: 0.713
Average epoch f1: 0.782
Epoch: 2 of 10


train batches: 100%|     | 326/326 [06:58<00:00,  1.28s/it, loss: 0.888]

Average epoch loss: 0.611

dev batches: 100%|     | 40/40 [00:15<00:00,  2.62it/s,  acc: 0.656, f1:
0.740]

Average epoch accuracy: 0.672
Average epoch f1: 0.751
Epoch: 3 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.020]

Average epoch loss: 0.296

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.655, f1:
0.739]

Average epoch accuracy: 0.658
Average epoch f1: 0.741
Epoch: 4 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.017]

Average epoch loss: 0.226

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.647, f1:
0.732]

Average epoch accuracy: 0.652
Average epoch f1: 0.736
Epoch: 5 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.008]

Average epoch loss: 0.158

dev batches: 100%|     | 40/40 [00:15<00:00,  2.61it/s,  acc: 0.652, f1:
0.735]

Average epoch accuracy: 0.650
Average epoch f1: 0.734
Epoch: 6 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.012]
```

```
Average epoch loss: 0.153

dev batches: 100%|     | 40/40 [00:15<00:00,  2.62it/s,  acc: 0.645, f1:
0.729]

Average epoch accuracy: 0.649
Average epoch f1: 0.732
Epoch: 7 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.005]

Average epoch loss: 0.120

dev batches: 100%|     | 40/40 [00:15<00:00,  2.62it/s,  acc: 0.640, f1:
0.725]

Average epoch accuracy: 0.643
Average epoch f1: 0.728
Epoch: 8 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.010]

Average epoch loss: 0.070

dev batches: 100%|     | 40/40 [00:15<00:00,  2.62it/s,  acc: 0.640, f1:
0.726]

Average epoch accuracy: 0.640
Average epoch f1: 0.726
Epoch: 9 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.004]

Average epoch loss: 0.077

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.633, f1:
0.721]

Average epoch accuracy: 0.636
Average epoch f1: 0.723
Epoch: 10 of 10


train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.004]

Average epoch loss: 0.099

dev batches: 100%|     | 40/40 [00:15<00:00,  2.62it/s,  acc: 0.637, f1:
0.723]

Average epoch accuracy: 0.635
Average epoch f1: 0.722
Done!
2023-05-08 03:01:46 model_05_cross_encoder_retrieval:INFO
```

```
run_best_epoch: 1, run_best_acc: 0.713405191898346, run_best_f1:
0.7815702557563782

2023-05-08 03:01:46 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-08 03:01:46 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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 03:01:48 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-08 03:01:48 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 10, 'n_neg_samples': 3, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 13153.53it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 326223.64it/s]

generated dataset n=953
Epoch: 1 of 10
```

```
train batches:  68%|       | 222/326 [04:46<02:14,  1.29s/it, loss: 0.774]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:59<00:00,  1.29s/it, loss: 0.834]

Average epoch loss: 1.561

dev batches: 100%|      | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.616, f1:
0.711]

Average epoch accuracy: 0.630
Average epoch f1: 0.721
Epoch: 2 of 10


train batches:  10%|          | 32/326 [00:41<06:19,  1.29s/it, loss: 0.558]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:58<00:00,  1.28s/it, loss: 0.915]

Average epoch loss: 0.542

dev batches: 100%|      | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.616, f1:
0.710]

Average epoch accuracy: 0.619
Average epoch f1: 0.713
Epoch: 3 of 10


train batches:  85%|      | 278/326 [05:57<01:01,  1.29s/it, loss: 0.039]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:58<00:00,  1.28s/it, loss: 0.104]

Average epoch loss: 0.261

dev batches: 100%|      | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.593, f1:
0.700]

Average epoch accuracy: 0.603
Average epoch f1: 0.705
Epoch: 4 of 10


train batches:  51%|        | 166/326 [03:33<03:26,  1.29s/it, loss: 0.008]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
```

```
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:58<00:00,  1.28s/it, loss: 0.096]

Average epoch loss: 0.150

dev batches: 100%|      | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.599, f1:
0.703]

Average epoch accuracy: 0.597
Average epoch f1: 0.703
Epoch: 5 of 10


train batches:  77%|       | 252/326 [05:24<01:35,  1.29s/it, loss: 0.025]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:58<00:00,  1.28s/it, loss: 0.304]

Average epoch loss: 0.155

dev batches: 100%|      | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.586, f1:
0.698]

Average epoch accuracy: 0.592
Average epoch f1: 0.701
Epoch: 6 of 10


train batches:  66%|       | 216/326 [04:38<02:21,  1.29s/it, loss: 0.049]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:59<00:00,  1.29s/it, loss: 0.048]

Average epoch loss: 0.135

dev batches: 100%|      | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.585, f1:
0.697]

Average epoch accuracy: 0.585
Average epoch f1: 0.698
Epoch: 7 of 10


train batches:   6%|          | 20/326 [00:25<06:35,  1.29s/it, loss: 0.002]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:59<00:00,  1.29s/it, loss: 0.026]

Average epoch loss: 0.108

dev batches: 100%|      | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.577, f1:
0.695]
```

```
Average epoch accuracy: 0.581
Average epoch f1: 0.696
Epoch: 8 of 10


train batches:  60%|       | 197/326 [04:13<02:46,  1.29s/it, loss: 0.004]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.002]

Average epoch loss: 0.118

dev batches: 100%|     | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.571, f1:
0.691]

Average epoch accuracy: 0.574
Average epoch f1: 0.693
Epoch: 9 of 10


train batches:   3%|         | 9/326 [00:11<06:50,  1.30s/it, loss: 0.006]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.046]

Average epoch loss: 0.078

dev batches: 100%|     | 40/40 [00:15<00:00,  2.63it/s,  acc: 0.564, f1:
0.688]

Average epoch accuracy: 0.567
Average epoch f1: 0.690
Epoch: 10 of 10


train batches:  89%|      | 289/326 [06:12<00:47,  1.29s/it, loss: 0.009]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:59<00:00,  1.29s/it, loss: 0.036]

Average epoch loss: 0.064

dev batches: 100%|     | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.559, f1:
0.686]

Average epoch accuracy: 0.561
Average epoch f1: 0.687
Done!
2023-05-08 04:14:13 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.6302266120910645, run_best_f1:
0.7211500406265259
```

```
2023-05-08 04:14:13 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-08 04:14:13 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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 04:14:25 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-08 04:14:25 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 10, 'n_neg_samples': 3, 'normalize_text': False,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 187056.67it/s]

generated dataset n=7806
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 329048.81it/s]

generated dataset n=953
Epoch: 1 of 10
```

```
train batches:  67%|       | 218/326 [04:41<02:19,  1.29s/it, loss: 1.562]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:59<00:00,  1.29s/it, loss: 1.918]

Average epoch loss: 1.530

dev batches: 100%|      | 40/40 [00:15<00:00,  2.66it/s,  acc: 0.619, f1:
0.723]

Average epoch accuracy: 0.613
Average epoch f1: 0.722
Epoch: 2 of 10


train batches:  32%|        | 105/326 [02:15<04:44,  1.29s/it, loss: 0.249]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:58<00:00,  1.28s/it, loss: 0.113]

Average epoch loss: 0.564

dev batches: 100%|      | 40/40 [00:15<00:00,  2.65it/s,  acc: 0.610, f1:
0.714]

Average epoch accuracy: 0.615
Average epoch f1: 0.719
Epoch: 3 of 10


train batches:   1%|          | 3/326 [00:03<06:57,  1.29s/it, loss: 0.046]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:58<00:00,  1.28s/it, loss: 0.417]

Average epoch loss: 0.269

dev batches: 100%|      | 40/40 [00:15<00:00,  2.65it/s,  acc: 0.605, f1:
0.713]

Average epoch accuracy: 0.607
Average epoch f1: 0.714
Epoch: 4 of 10


train batches:  51%|        | 167/326 [03:34<03:24,  1.28s/it, loss: 0.095]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|      | 326/326 [06:57<00:00,  1.28s/it, loss: 0.004]
```

```
Average epoch loss: 0.185

dev batches: 100%|     | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.603, f1:
0.710]

Average epoch accuracy: 0.604
Average epoch f1: 0.711
Epoch: 5 of 10


train batches:  34%|        | 112/326 [02:23<04:35,  1.29s/it, loss: 0.008]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.003]

Average epoch loss: 0.118

dev batches: 100%|     | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.593, f1:
0.705]

Average epoch accuracy: 0.599
Average epoch f1: 0.708
Epoch: 6 of 10


train batches:  93%|     | 303/326 [06:29<00:29,  1.29s/it, loss: 0.545]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.006]

Average epoch loss: 0.123

dev batches: 100%|     | 40/40 [00:15<00:00,  2.65it/s,  acc: 0.587, f1:
0.701]

Average epoch accuracy: 0.591
Average epoch f1: 0.704
Epoch: 7 of 10


train batches:  26%|        | 85/326 [01:49<05:09,  1.28s/it, loss: 0.002]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.002]

Average epoch loss: 0.091

dev batches: 100%|     | 40/40 [00:15<00:00,  2.64it/s,  acc: 0.589, f1:
0.701]

Average epoch accuracy: 0.588
Average epoch f1: 0.701
```

```
Epoch: 8 of 10


train batches:  98%|     | 321/326 [06:52<00:06,  1.29s/it, loss: 0.041]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.054]

Average epoch loss: 0.118

dev batches: 100%|     | 40/40 [00:15<00:00,  2.65it/s,  acc: 0.585, f1:
0.698]

Average epoch accuracy: 0.587
Average epoch f1: 0.700
Epoch: 9 of 10


train batches:  46%|       | 151/326 [03:13<03:45,  1.29s/it, loss: 0.010]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:57<00:00,  1.28s/it, loss: 0.003]

Average epoch loss: 0.109

dev batches: 100%|     | 40/40 [00:15<00:00,  2.65it/s,  acc: 0.579, f1:
0.695]

Average epoch accuracy: 0.582
Average epoch f1: 0.697
Epoch: 10 of 10


train batches:  86%|     | 279/326 [05:58<01:00,  1.29s/it, loss: 0.054]Be
aware, overflowing tokens are not returned for the setting you have chosen, i.e.
sequence pairs with the 'longest_first' truncation strategy. So the returned
list will always be empty even if some tokens have been removed.
train batches: 100%|     | 326/326 [06:58<00:00,  1.28s/it, loss: 0.002]

Average epoch loss: 0.098

dev batches: 100%|     | 40/40 [00:15<00:00,  2.65it/s,  acc: 0.571, f1:
0.692]

Average epoch accuracy: 0.575
Average epoch f1: 0.694
Done!
2023-05-08 05:26:39 model_05_cross_encoder_retrieval:INFO
run_best_epoch: 1, run_best_acc: 0.6148208379745483, run_best_f1:
0.7218205332756042

2023-05-08 05:26:39 model_05_cross_encoder_retrieval:INFO
```

```
== CURRENT BEST F1: 0.8339534997940063

2023-05-08 05:26:39 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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 05:26:40 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-08 05:26:40 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 10, 'n_neg_samples': 5, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.01}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 156358.50it/s]

generated dataset n=10260
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 275446.83it/s]

generated dataset n=1261
Epoch: 1 of 10


train batches: 100%|     | 428/428 [09:09<00:00,  1.28s/it, loss: 0.708]

Average epoch loss: 1.595
```

```
dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.761, f1:
0.743]

Average epoch accuracy: 0.762
Average epoch f1: 0.751
Epoch: 2 of 10


train batches: 100%|     | 428/428 [09:06<00:00,  1.28s/it, loss: 0.145]

Average epoch loss: 0.585

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.719, f1:
0.707]

Average epoch accuracy: 0.735
Average epoch f1: 0.722
Epoch: 3 of 10


train batches: 100%|     | 428/428 [09:08<00:00,  1.28s/it, loss: 0.045]

Average epoch loss: 0.317

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.695, f1:
0.692]

Average epoch accuracy: 0.706
Average epoch f1: 0.700
Epoch: 4 of 10


train batches: 100%|     | 428/428 [09:06<00:00,  1.28s/it, loss: 0.013]

Average epoch loss: 0.174

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.701, f1:
0.695]

Average epoch accuracy: 0.699
Average epoch f1: 0.695
Epoch: 5 of 10


train batches: 100%|     | 428/428 [09:08<00:00,  1.28s/it, loss: 0.068]

Average epoch loss: 0.204

dev batches: 100%|     | 53/53 [00:20<00:00,  2.62it/s,  acc: 0.691, f1:
0.688]

Average epoch accuracy: 0.697
Average epoch f1: 0.693
Epoch: 6 of 10


train batches: 100%|     | 428/428 [09:05<00:00,  1.27s/it, loss: 0.003]
```

```
Average epoch loss: 0.144

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.672, f1:
0.676]

Average epoch accuracy: 0.681
Average epoch f1: 0.682
Epoch: 7 of 10


train batches: 100%|     | 428/428 [09:08<00:00,  1.28s/it, loss: 0.004]

Average epoch loss: 0.109

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.662, f1:
0.672]

Average epoch accuracy: 0.667
Average epoch f1: 0.674
Epoch: 8 of 10


train batches: 100%|     | 428/428 [09:05<00:00,  1.28s/it, loss: 0.006]

Average epoch loss: 0.095

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.654, f1:
0.667]

Average epoch accuracy: 0.659
Average epoch f1: 0.670
Epoch: 9 of 10


train batches: 100%|     | 428/428 [09:08<00:00,  1.28s/it, loss: 0.002]

Average epoch loss: 0.091

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.645, f1:
0.661]

Average epoch accuracy: 0.650
Average epoch f1: 0.664
Epoch: 10 of 10


train batches: 100%|     | 428/428 [09:06<00:00,  1.28s/it, loss: 0.001]

Average epoch loss: 0.065

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.641, f1:
0.659]

Average epoch accuracy: 0.644
Average epoch f1: 0.660
Done!
2023-05-08 07:01:16 model_05_cross_encoder_retrieval:INFO
```

```
run_best_epoch: 1, run_best_acc: 0.7624387145042419, run_best_f1:
0.751181960105896

2023-05-08 07:01:16 model_05_cross_encoder_retrieval:INFO
== CURRENT BEST F1: 0.8339534997940063

2023-05-08 07:01:16 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 5, 'n_neg_samples': 3, '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.predictions.transform.dense.bias',
'cls.seq_relationship.bias', 'cls.predictions.transform.dense.weight',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.decoder.weight', 'cls.seq_relationship.weight',
'cls.predictions.transform.LayerNorm.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 07:01:18 model_05_cross_encoder_retrieval:INFO
== RUN

2023-05-08 07:01:18 model_05_cross_encoder_retrieval:INFO
{'batch_size': 24, 'claims_paths':
[PosixPath('/Users/johnsonzhou/git/comp90042-project/data/train-claims.json')],
'claims_shortlist_paths': [PosixPath('result/pipeline/shortlisting_v2/train_retr
ieved_evidences_max_500_no_rel.json')], 'dropout': None, 'lr': 5e-05,
'max_length': 512, 'n_epochs': 10, 'n_neg_samples': 5, 'normalize_text': True,
'warmup': 0.1, 'weight_decay': 0.02}

Torch device is 'mps'

claims: 100%|     | 1228/1228 [00:00<00:00, 268358.53it/s]

generated dataset n=10260
Torch device is 'mps'

claims: 100%|     | 154/154 [00:00<00:00, 255305.46it/s]

generated dataset n=1261
Epoch: 1 of 10
```

```
train batches: 100%|     | 428/428 [09:09<00:00,  1.28s/it, loss: 1.404]

Average epoch loss: 1.783

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.593, f1:
0.647]

Average epoch accuracy: 0.594
Average epoch f1: 0.651
Epoch: 2 of 10


train batches: 100%|     | 428/428 [09:11<00:00,  1.29s/it, loss: 0.095]

Average epoch loss: 0.617

dev batches: 100%|     | 53/53 [00:20<00:00,  2.59it/s,  acc: 0.551, f1:
0.619]

Average epoch accuracy: 0.568
Average epoch f1: 0.631
Epoch: 3 of 10


train batches: 100%|     | 428/428 [09:18<00:00,  1.31s/it, loss: 0.020]

Average epoch loss: 0.285

dev batches: 100%|     | 53/53 [00:20<00:00,  2.62it/s,  acc: 0.587, f1:
0.636]

Average epoch accuracy: 0.571
Average epoch f1: 0.629
Epoch: 4 of 10


train batches: 100%|     | 428/428 [09:10<00:00,  1.29s/it, loss: 0.085]

Average epoch loss: 0.222

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.597, f1:
0.639]

Average epoch accuracy: 0.592
Average epoch f1: 0.638
Epoch: 5 of 10


train batches: 100%|     | 428/428 [09:08<00:00,  1.28s/it, loss: 0.296]

Average epoch loss: 0.142

dev batches: 100%|     | 53/53 [00:20<00:00,  2.62it/s,  acc: 0.602, f1:
0.641]
```

```
Average epoch accuracy: 0.600
Average epoch f1: 0.641
Epoch: 6 of 10


train batches: 100%|     | 428/428 [09:09<00:00,  1.28s/it, loss: 0.032]

Average epoch loss: 0.127

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.589, f1:
0.634]

Average epoch accuracy: 0.595
Average epoch f1: 0.638
Epoch: 7 of 10


train batches: 100%|     | 428/428 [09:07<00:00,  1.28s/it, loss: 0.003]

Average epoch loss: 0.127

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.588, f1:
0.632]

Average epoch accuracy: 0.589
Average epoch f1: 0.634
Epoch: 8 of 10


train batches: 100%|     | 428/428 [09:09<00:00,  1.28s/it, loss: 0.023]

Average epoch loss: 0.100

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.599, f1:
0.638]

Average epoch accuracy: 0.594
Average epoch f1: 0.636
Epoch: 9 of 10


train batches: 100%|     | 428/428 [09:08<00:00,  1.28s/it, loss: 0.016]

Average epoch loss: 0.108

dev batches: 100%|     | 53/53 [00:20<00:00,  2.63it/s,  acc: 0.589, f1:
0.633]

Average epoch accuracy: 0.594
Average epoch f1: 0.635
Epoch: 10 of 10


train batches:  29%|        | 126/428 [02:43<06:33,  1.30s/it, loss: 0.010]
```