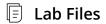


***	Navigate
\Box	INAVIGACE





```
0.295800
          300
          350
                  0.413400
         TrainOutput(global step=375, training loss=0.6163130976359049, metrics={'train r
         untime': 13.5447, 'train_samples_per_second': 27.686, 'total_flos': 310585476240
         00, 'epoch': 3.0})
In [31]:
          trainer.evaluate(test_ds)
                                                 [125/125 00:01]
Out[31]: {'eval_loss': 0.32361623644828796,
           'eval f1 start': 0.7895575553055868,
          'eval f1 end': 0.7883126550868487,
           'eval runtime': 1.9534,
```

Now it is time to ask your PyTorch model a question!

'eval_samples_per_second': 511.926,

'epoch': 3.0}

- Before testing your model with a question, you can tell PyTorch to send your model and inputs to the GPU if your machine has one, or the CPU if it does not.
- You can then proceed to tokenize your input and create PyTorch tensors and send them to your device.
- The rest of the pipeline is relatively similar to the one you implemented for TensorFlow.

```
In [32]:
          import torch
          device = torch.device('cuda') if torch.cuda.is available() else torch.device('cp')
          pytorch model.to(device)
          question, text = 'What is east of the hallway?', 'The kitchen is east of the hall
          input dict = tokenizer(text, question, return tensors='pt')
          input ids = input_dict['input_ids'].to(device)
          attention mask = input dict['attention mask'].to(device)
          outputs = pytorch model(input ids, attention mask=attention mask)
          start logits = outputs[0]
          end logits = outputs[1]
          all tokens = tokenizer.convert ids to tokens(input dict["input ids"].numpy()[0])
          answer = ' '.join(all tokens[torch.argmax(start logits, 1)[0]: torch.argmax(end
          print(question, answer.capitalize())
```

What is east of the hallway? Kitchen

Congratulations!