



300	0.295800
350	0.413400

```
Out[30]: 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,
'eval_samples_per_second': 511.926,
'epoch': 3.0}
```

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

- 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!