In [ ]:
```
!pip install transformers torch
```

In [ ]:
```
from transformers import BertTokenizer, BertForSequenceClassification
import torch
```

In [ ]:
```
model_name = 'bert-base-uncased'
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertForSequenceClassification.from_pretrained(model_name)
```

In [ ]:
```
from datasets import load_dataset
dataset = load_dataset('glue', 'sst2')
```

In [ ]:
```
def tokenize_data(example):
    return tokenizer(example['sentence'], padding='max_length', truncation=True)

tokenized_dataset = dataset.map(tokenize_data, batched=True)
```

In [ ]:
```
from transformers import Trainer, TrainingArguments

training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=3,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
```

In [ ]:
```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset['train'],
    eval_dataset=tokenized_dataset['validation'],
)
```

In [ ]:
```
trainer.train()
```

In [ ]:
```
results = trainer.evaluate()
print(results)
```