NLP Final Project

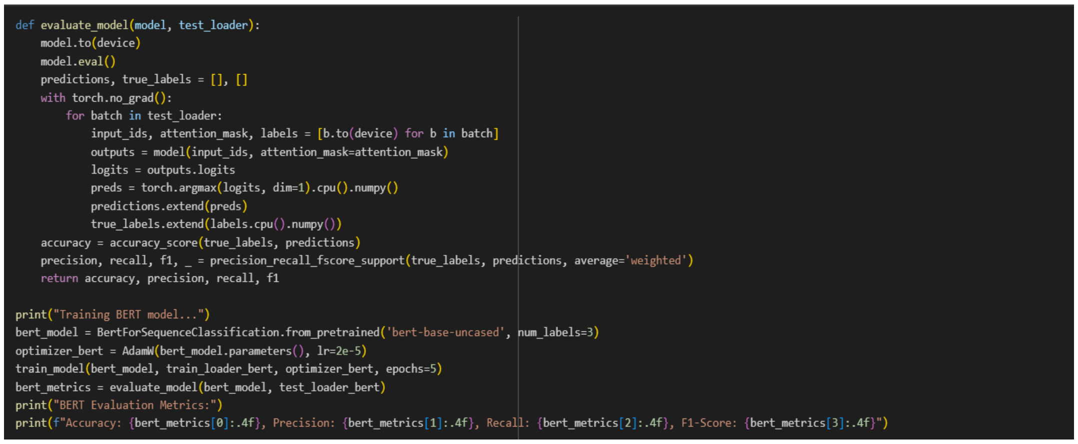
Sara Ehab 202202146, Menna Sharaf 202201707 ,

Salma Rramy 202201759 , Shahid Tarek 202202263

PROJECT PURPOSE

This project aims to implement an automatic review analysis system using sentiment analysis techniques. The main goal is to evaluate and compare the performance of two powerful transformer-based models (**BERT** and **RoBERTa)**.

BERT Model



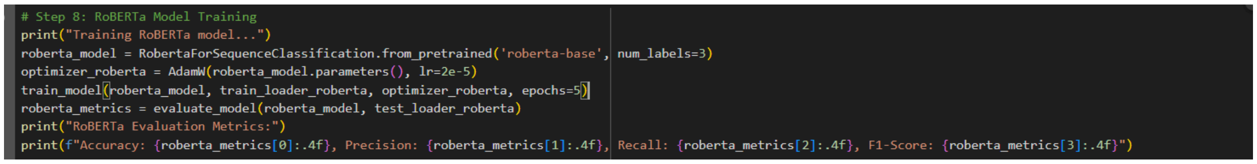
```
def evaluate_model(model, test_loader):
    model.to(device)
    model.eval()
    predictions, true_labels = [], []
    with torch.no_grad():
        for batch in test_loader:
            input_ids, attention_mask, labels = [b.to(device) for b in batch]
            outputs = model(input_ids, attention_mask=attention_mask)
            logits = outputs.logits
            preds = torch.argmax(logits, dim=1).cpu().numpy()
            predictions.extend(preds)
            true_labels.extend(labels.cpu().numpy())
    accuracy = accuracy_score(true_labels, predictions)
    precision, recall, f1, _ = precision_recall_fscore_support(true_labels, predictions, average='weighted')
    return accuracy, precision, recall, f1

print("Training BERT model...")
bert_model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=3)
optimizer_bert = AdamW(bert_model.parameters(), lr=2e-5)
train_model(bert_model, train_loader_bert, optimizer_bert, epochs=5)
bert_metrics = evaluate_model(bert_model, test_loader_bert)
print("BERT Evaluation Metrics:")
print(f"Accuracy: {bert_metrics[0]:.4f}, Precision: {bert_metrics[1]:.4f}, Recall: {bert_metrics[2]:.4f}, F1-Score: {bert_metrics[3]:.4f}")
```

1. Accuracy: 0.6933
2. Precision :0.7078
3. Recall: 0.6933
4. F1-score : 0.6951

Analysis: the BERT model performed well, but there's still room for improvement, especially in making even more accurate predictions.

RoBerTa Model



```
# Step 8: RoBERTa Model Training
print("Training RoBERTa model...")
roberta_model = RobertaForSequenceClassification.from_pretrained('roberta-base', num_labels=3)
optimizer_roberta = AdamW(roberta_model.parameters(), lr=2e-5)
train_model(roberta_model, train_loader_roberta, optimizer_roberta, epochs=5)
roberta_metrics = evaluate_model(roberta_model, test_loader_roberta)
print("RoBERTa Evaluation Metrics:")
print(f"Accuracy: {roberta_metrics[0]:.4f}, Precision: {roberta_metrics[1]:.4f}, Recall: {roberta_metrics[2]:.4f}, F1-Score: {roberta_metrics[3]:.4f}")
```

1. Accuracy : 0.7033
2. Precision : 0.6996
3. Recall: 0.7033
4. F1-score :0.6986

Analysis : The RoBERTa model achieved an accuracy of 70.33%, slightly higher than BERT, meaning it correctly predicted the sentiment in just over 7 out of 10 reviews.

Comparison

Model comparison

Bert-Accuracy : 0.6933 , F1-score :0.6951

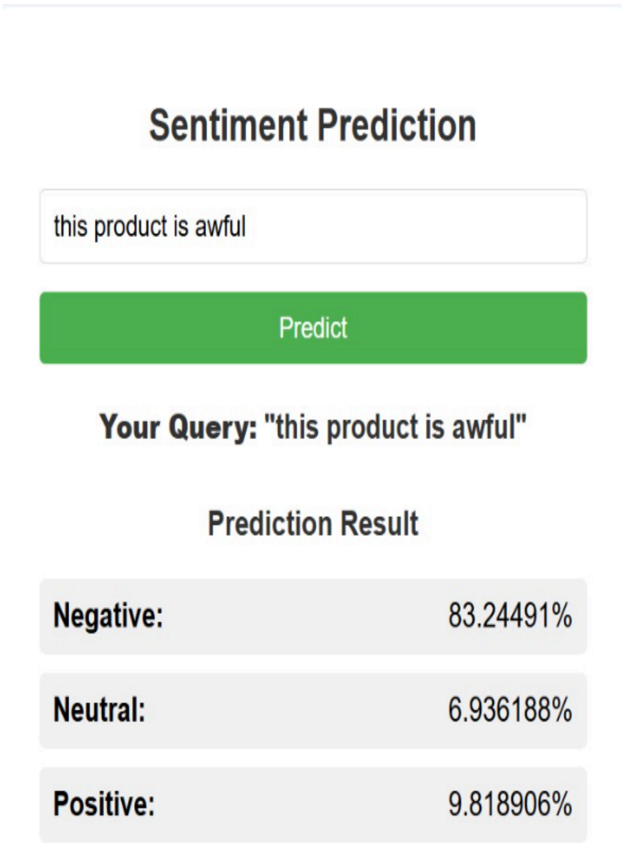
Roberta -Accuracy :0.7033 , F1-score : 0.6986

RoBERTa slightly outperformed BERT, achieving higher accuracy (70.33% vs. 69.33%) and a better F1-score (69.86% vs. 69.51%), making it the more effective model in this comparison.

Gui

The model can predict correctly to which class query belong too

QUERY1 :



Sentiment Prediction

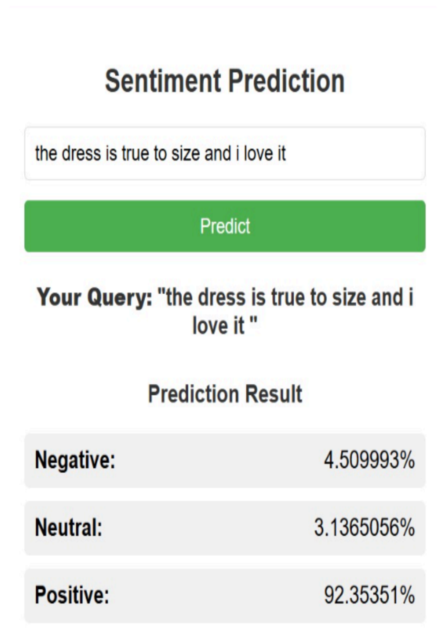
this product is awful

Your Query: "this product is awful"

Prediction Result

Negative:	83.24491%
Neutral:	6.936188%
Positive:	9.818906%

QUERY 2:


Sentiment Prediction

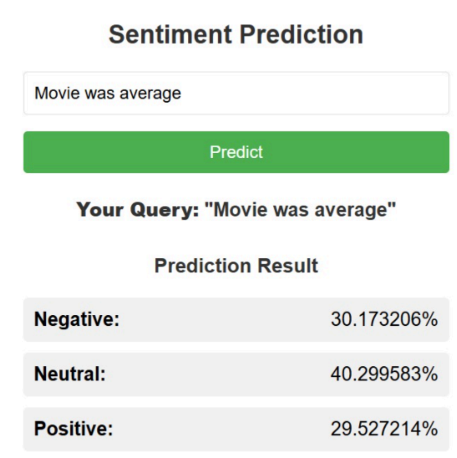
the dress is true to size and i love it

Your Query: "the dress is true to size and i love it "

Prediction Result

Negative:	4.509993%
Neutral:	3.1365056%
Positive:	92.35351%

QUERY3:


Sentiment Prediction

Movie was average

Your Query: "Movie was average"

Prediction Result

Negative:	30.173206%
Neutral:	40.299583%
Positive:	29.527214%