Báo Cáo Bert và Distilbert

1. **Tạo bộ dữ liệu train, valid, test:**

* 60% dữ liệu đầu cho training.
* 20% dữ liệu tiếp theo cho validation.
* 20% dữ liệu cuối cùng cho testing.

1. **Xây dựng mô hình với số nhãn là 2:**

* Mô hình Bert: 'bert-base-uncased'.
* Mô hình XLNet: 'xlnet-base-cased'.

1. **Class Tokenize:**

class Tokenizer:

def \_\_init\_\_(self, model\_name, max\_length=128):

if 'bert' in model\_name:

self.tokenizer = BertTokenizerFast.from\_pretrained(model\_name)

 elif 'xlnet' in model\_name:

      self.tokenizer = XLNetTokenizerFast.from\_pretrained(model\_name)

  self.max\_length = max\_length

def tokenize\_function(self, examples):

return self.tokenizer(

  examples["text"],

      truncation=True,

      padding="max\_length",

      max\_length=self.max\_length

  )

1. **Thiết lập Hyperparameter cho training:**

* Output\_dir=’./bert\_results’. # Lưu kết quả huấn luyện vào file bert\_results.
* per\_device\_train\_batch\_size=8.
* per\_device\_eval\_batch\_size=8. # Thiết lập kích thước batch cho tập huấn luyện và tập đánh giá.
* num\_train\_epochs=1 # Số lần mô hình huấn luyện trên toàn bộ tập dữ liệu.
* weight\_decay=0.01 # Áp dụng kỹ thuật điều chuẩn L2 để giúp mô hình tránh overfitting.
* evaluation\_strategy="epoch" # Tần suất đánh giá mô hình.
* save\_strategy="epoch" # Tần suất lưu loss, learning\_rate, grad\_norm, trọng số của mô hình.

1. **Huấn luyện mô hình:**

* Sử dụng các thông số như accuracy, f1-score, recall, precision-score để đánh giá.

|  |  |  |
| --- | --- | --- |
| Chỉ số | Bert | Distilbert |
| Thời gian Training | ~13 phút (783s) | ~7.8 phút (471s) |
| Thời gian Đánh giá | ~1.5 phút (88.7s) | ~53.5s |
| Accuracy | 94.26% | 93.81% |
| F1-Score (Weighted) | **0.9427** | **0.9382** |
| Precision (Weighted) | 0.943 | 0.9389 |
| Recall (Weighted) | 0.9426 | 0.9381 |
| Loss trên tập Train | 0.242 | 0.2334 |
| Loss trên tập Validation | 0.1549 | 0.1641 |
| Loss trên tập Test | 0.1625 | 0.17 |
| Confusion Matrix | [[7174, 464],  [323, 5759]] | [[7095, 543],  [306, 5776]] |

1. **Đánh giá mô hình:**
2. **Kết luận:**

* Mô hình Bert cho ra các thông số đánh giá tốt hơn nhưng đòi hỏi tài nguyên tính toán và thời gian huấn luyện đáng kể (13 phút cho 1 epoch).
* Mô hình Distilbert có thời gian huấn luyện và đánh giá **nhanh hơn khoảng 66%** và kích thước nhỏ hơn 40%, nó mang lại một sự đánh đổi gần như hoàn hảo giữa tốc độ và độ chính xác.
* Nếu mục tiêu là đạt được **độ chính xác tối đa** mà không quá quan tâm đến chi phí tính toán, **bert-base-uncased** là lựa chọn tốt nhất.
* Tuy nhiên, trong các ứng dụng thực tế nơi tốc độ inference và chi phí triển khai là yếu tố quan trọng, **distilbert-base-uncased** là một lựa chọn vượt trội. Nó chỉ hy sinh một phần rất nhỏ (chưa đến 1%) về hiệu năng để đổi lấy tốc độ nhanh hơn đáng kể và yêu cầu tài nguyên ít hơn.

Đối với bài toán này, sự đánh đổi của Distilbert là hoàn toàn chấp nhận được và nó được xem là mô hình hiệu quả hơn về tổng thể.