1. تحميل المكتبات

Python

import pandas as pd

import numpy as np

import tensorflow as tf

from transformers import AutoTokenizer, TFAutoModelForSequenceClassification

from sklearn.preprocessing import LabelEncoder

from tensorflow.keras.optimizers.legacy import Adam

from joblib import dump

هذه الخطوة تحمل المكتبات اللازمة للكود. المكتبات هي:

pandas لتعامل مع البيانات.

numpy للعمليات الحسابية.

tensorflow لإنشاء النموذج.

transformers لاستخدام نموذج BERT.

sklearn لتحويل التصنيفات إلى أرقام.

joblib لحفظ النموذج.

2. تحميل القاموس

Python

dictionary\_file\_path = "C:\\py\\Last\\split\_words.xlsx" # مسار ملف القاموس

dictionary\_data = pd.read\_excel(dictionary\_file\_path)

هذه الخطوة تحمل البيانات من ملف Excel. الملف يجب أن يحتوي على عمودين: "term" و "classification".

3. استخراج الكلمات والتصنيفات

Python

terms = dictionary\_data['term'].astype(str).tolist()

categories = dictionary\_data['classification'].astype(str).tolist()

هذه الخطوة تستخرج الكلمات والتصنيفات من البيانات.

4. تحويل التصنيفات إلى أرقام

Python

label\_encoder = LabelEncoder()

y = label\_encoder.fit\_transform(categories)

n\_classes = len(label\_encoder.classes\_)

هذه الخطوة تحول التصنيفات إلى أرقام باستخدام LabelEncoder. هذا يسهل على النموذج التعلم.

5. تحميل Tokenizer الخاص بـ BERT

Python

tokenizer = AutoTokenizer.from\_pretrained("bert-base-uncased")

X\_encoded = tokenizer(terms, padding=True, truncation=True, return\_tensors="tf", max\_length=128)

هذه الخطوة تحمل Tokenizer الخاص بـ BERT. هذا يتحول النصوص إلى تنسيق يمكن للنموذج فهمه.

6. تحميل نموذج BERT للتصنيف

Python

model = TFAutoModelForSequenceClassification.from\_pretrained("bert-base-uncased", num\_labels=n\_classes)

هذه الخطوة تحمل نموذج BERT للتصنيف.

7. تجميع النموذج

Python

optimizer = Adam(learning\_rate=2e-5)

model.compile(optimizer=optimizer, loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True), metrics=['accuracy'])

هذه الخطوة تجميع النموذج. النموذج يستخدم خوارزمية Adam لتحسين الأوزان، ووظيفة الخسارة هي SparseCategoricalCrossentropy.

8. تدريب النموذج

Python

model.fit(dict(X\_encoded), y, epochs=3, batch\_size=16, validation\_split=0.2)

هذه الخطوة تدريب النموذج. النموذج يتدرب على البيانات لمدة 3 دورات، مع حجم=batch 16، و20% من البيانات

تستخدم للتحقق.

9. حفظ النموذج والأدوات المساعدة

Python

model\_file\_path = "C:\\py\\Last\\bert\_model"

tokenizer\_file\_path = "C:\\py\\Last\\bert\_tokenizer.joblib"

label\_encoder\_file\_path = "C:\\py\\Last\\label\_encoder.joblib"

model.save\_pretrained(model\_file\_path)

dump(tokenizer, tokenizer\_file\_path)

dump(label\_encoder, label\_encoder\_file\_path)

print(f"تم حفظ النموذج في: {model\_file\_path}")

print(f"تم حفظ الـ Tokenizer في: {tokenizer\_file\_path}")

print(f"تم حفظ الـ Label Encoder في: {label\_encoder\_file\_path}")