Manual: How to Use the Bug Report Classification Tool

This project contains two Python scripts for classifying whether GitHub bug reports are performance-related:

- br_classification.py: The baseline method using TF-IDF + Naive Bayes.
- bert_classification.py: The proposed method using a pretrained BERT model.

Requirements

Make sure to install all necessary dependencies using the provided requirements.txt. You can install them using:

```
pip install -r requirements.txt
```

For first-time use of nltk, download stopwords by running:

```
import nltk
nltk.download('stopwords')
```

Input Data

This tool expects input CSV files for each project, named as:

```
pytorch.csv
tensorflow.csv
keras.csv
incubator-mxnet.csv
caffe.csv
```

These files must be placed in the same directory as the script or you should modify the path accordingly.

How to Run

Each script includes a line to select the project name. Before running, open the script and locate the following lines near the top:

```
# Choose the project (options: 'pytorch', 'tensorflow', 'keras', 'incubator-mxnet', 'caffe')
project = 'keras'
path = f'{project}.csv'
```

You should change 'keras' to the name of the project you want to run, for example:

```
project = 'pytorch'
path = f'{project}.csv'
```

Then save the file and run:

Run baseline method:

```
python br_classification.py
```

Run BERT method:

```
python bert_classification.py
```

Output

Both scripts will:

- Print performance metrics to the console (Accuracy, Precision, Recall, F1, AUC).
- Save results to a CSV file for future analysis.

Notes

- The BERT script uses bert-base-uncased from HuggingFace and does not fine-tune BERT for efficiency. Only the classifier head is trained.
- If you wish to enable full fine-tuning or modify hyperparameters, you can edit bert_classification.py accordingly.
- For small datasets (e.g., caffe.csv), performance may be lower due to class imbalance and limited samples.