A. Artifact Appendix

A.1 Abstract

The artifact contains the source code and the data used for the work titled: "Can citations tell us about a paper's reproducibility? A case study of machine learning papers". The project requires Python GPU-based processing capabilities, TensorFlow, Keras and PyTorch frameworks. Users can reproduce the results in Table 2, Table 3, Table 4, Figure 3, and Figure 4 using the Jupyter Notebooks shared through GitHub and Zenodo.

A.2 Artifact check-list (meta-information)

- **Compilation:** Python 3.10.8 [Linux GCC 11.2.0]
- **Model:** The code-base retrieves and downloads five HuggingFace models (∼3GB) and DistilBERT from their repositories as needed during execution.
- **Dataset:** All the necessary data is included in 'data' directory
- **Run-time environment:** Artifact is not OS-specific, but Linux is preferred. Tested on Linux 4.18.0-513.9.1.el8_9.x86_64. Using Python VENV would alleviate the necessity for root access. Code-base is successfully re-executed with Python 3.10.8, Tensorflow-2.16.1, Keras-3.1.1 and PyTroch-1.13.0.
- **Hardware:** Minimum 16 GB GPU required (Tested with Tesla T4 and Tesla V100)
- **Metrics:** Mean Average Precision (mAP), Mean Average Recall(mAR) and Mean Average F1(mAF1) is used to evaluate model performances.
- **Output:** Expected results are included in the article as Table 2, Table 3, Table 4, Figure 3, and Figure 4
- **Experiments:** Shared 'README.md' file explains the necessary steps required. All experiments are inlcuded as code snippets through Jupyter Notebooks available in the 'notebooks' directory. Maximum allowable variation for all model performance measures is $\pm 5\%$.
- **How much disk space required (approximately)?:** 10GB
- **How much time is needed to prepare workflow (approximately)?:** 1 hour
- **How much time is needed to complete experiments (approximately)?:** 6+ hours
- **Publicly available?:** Yes.
- **Code licenses (if publicly available)?:** MIT license
- **Data licenses (if publicly available)?:** Creative Commons Attribution 4.0 International
- **Workflow framework used?:** No
- **Archived (provide DOI)?:** 10.5281/zenodo.10895748

A.3 Description

A.3.1 How to access

To access the artifact, you can clone the publicly available GitHub repository https://github.com/lamps-lab/ccair-ai-reproducibility. The source-code is also available via 10.5281/zenodo.10895748. Once cloned, the initial repository size will be around ∼200MB

A.3.2 Hardware dependencies

- In Google colab - Tesla T4 16 GB
- Internal Cluster - Tesla V100-SXM2-16GB with 4 cores

A.3.3 Software dependencies

All the required dependencies included in the "requirements.txt" file. To prevent dependency conflicts, **refrain from manually installing TensorFlow and Keras**. When installing keras-nlp via requirements.txt, it will automatically download and install the appropriate TensorFlow and Keras versions. Artifact is tested on below python library versions.

- tensorflow−2.16.1
- keras−3.1.1
- keras-core−0.1.7
- keras-nlp−0.8.2
- torch−1.13.0
- transformers−4.39.2
- pandas−2.0.3
- ipykernel−6.29.3
- openpyxl−3.1.2
- numpy−1.24.3
- scikit-learn−1.3.1

A.3.4 Data sets

All data is available through GitHub `'https://github.com/lamps-lab/ccair-ai-reproducibility/tree/main/data'`

A.4 Installation

Steps to follow:

1. Clone the GitHub repository https://github.com/lamps-lab/ccair-ai-reproducibility
2. Create a python virtual environment `https://docs.python.org/3/library/venv.html`
3. Activate venv, navigate to the cloned repository and install the dependencies using requirements.txt file

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

4. Use either the available data in 'data' directory or create the datasets from scratch by following the steps in below jupyter notebooks in sequential order (available inside 'notebooks' directory).
   (a) `R_001_Creating_the_RS_superset.ipynb`
   (b) `R_001_Extract_Citing_Paper_Details_from_S2GA.ipynb`
   (c) `R_001_JSON_to_csv_contexts_conversion.ipynb`

   Note: If you are using the existing data in the 'data' directory, you can skip this step 4)

A.5 Experiment workflow

After the environment setup, execute the below jupyter notebooks in sequential order (available inside 'notebooks' directory).

1. `R_001_M1_to_M5_Sentiment_Analysis_models.ipynb`
   - This will generate the performance measures for the selected five open-source multiclass sentiment analysis models (Table 3)
2. `R_001_M6_3_class_sentiment_classification.ipynb`
   - This will custom train a multiclass DistilBert sentiment classifier and perform 5-fold cross validation for model evaluation. At the end of model evaluation this generates the predicted class labels {'negative','neutral','positive'} for all 41244 citation contexts (Table 4)
3. `R_001_M7_1_binary_classification_related_not_related.ipynb`
   - This will custom train a binary classifier and perform 5-fold cross validation for model evaluation. At the end of model evaluation this generates the predicted class labels {'related','not-related'} for all 41244 citation contexts. (Table 4)
4. `R_001_M7_2_binary_sentiment_classification.ipynb`

- This will custom train a binary classifier and perform 5-fold cross validation for model evaluation. At the end of model evaluation this generates the predicted class labels {'negative','positive'} for only reproducibility related citation contexts filtered from M7.1 (Table 4).

5. `R_001_Visualizations.ipynb`
    - This will parse all the data files created by previous notebooks and generate the results in Table 2, figure 3 and Figure 4.

A.6 Evaluation and expected results

Expected results are reported in the article as Table 2, Table 3, Table 4, Figure 3, and Figure 4 and these results will be generated by the successful execution of below notebooks.

1. `Table 2, Figure 3, and Figure 4`
    - `R_001_Visualizations.ipynb`
2. `Table 3`
    - `R_001_M1_to_M5_Sentiment_Analysis_***.ipynb`
3. `Table 4`
    - `R_001_M6_3_class_sentiment_cla***.ipynb`
    - `R_001_M7_1_binary_classification_re***.ipynb`
    - `R_001_M7_2_binary_sentiment_cla***.ipynb`