Team Name: Adacore
Team members Information: Amrit Prasad Phuyal & Dipendra Raj Panta
Private Leaderboard Score: 0.46694106893526294
Private Leaderboard Place: 14

Model used: LSTM (Long Short Term Memory) a version of RNN.
Tools used : Google colab(for online GPU) and Laptop to run locally
Training Time: 45 minutes and more
System specification:

- Windows 11 pro
- Nvidia gtx 1050 Ti
- Core i7 , 6 core
- 16 gb Ram
- Python 3.9.7 (64bit)
- Anaconda Python (for packages) (latest)
- Vscode with Jupyter extension (latest)

Steps to Run

- Create virtual environment
- Activate virtual environment
- Install dependencies (use requirement.txt file)
- **For GPU usage**
    - *(Local only)* Install Tensorflow , Tensorflow-gpu , Keras and Keras-gpu from Anaconda repository that will install necessary Library (cuDNN) and Nvidia Kernel (Cuda).
    - *(Google Colab)* no need to install these libraries but don't forget to select GPU as runtime.
- **Data preparation** (cleaning and stemming)
    - *(Local)* All Data has to be in "data" folder in your root directory.
    - *(Google Colab)* Data has to be in "data" folder in your root directory.and change the path accordingly in notebook.
    - Run prepare_data.py file or prepare_data.ipynb Notebook. It takes time to finish.
    - New files `train_clean.csv` and `test_clean.csv` will be created in "data" folder.
- **Training**
    - We will use `train_clean.csv` from data folder to train model
        - Run train.py file or train.ipynb Notebook. It takes time to finish.If your GPU supports increase the batch size for faster training.
    - In the process following files will be generated in root folder
        - `train_clean_counts_word.csv` >> To view no of words in each abstract and help to determine suitable value for MAX_SEQUENCE_LENGTH
        - `word_index.csv` >> To view index number assigned to each word
        - `word_counts.csv` >> To view the number of times a word is repeated and determine the dictionary length MAX_NB_WORDS

    - Training will take around an hour and trained model will be saved in `Saved_model` folder .
    - Accuracy and Loss plot is generated at last.
- **Prediction**
    - Import trained model from `Saved_model` folder
    - Run predict.py file or predict.ipynb Notebook.
    - New file `solution.csv` will be created in root folder.

Data Description

Datasets is available at
https://drive.google.com/drive/folders/1hOpQ2LpKixp3sj3CGv9700ETIp2tUkFR

Data for competition should be in the form of csv file at 'data' folder in root Directory

File	Decription
`train.csv`	Training dataset containing id, abstract, category and category_num
`validation.csv`	Validation dataset containing id, abstract, category and category_num
`test.csv`	The testing dataset contains id, abstract. The competitors are required to predict category_num
`labels.csv`	Map of the category to category_num
`sample.csv`	The sample solution file. The id column is the same as of `test.csv` category_num is to the predicted output.

The solution file should be named `solution.csv` and the format is the same as `sample.csv`