ADVANCE Labs - Cyberbullying Detection Lab

The development of this document is partially funded by the National Science Foundation's Security and Trustworthy Cyberspace Education, (SaTC-EDU) program under Award No. 2114920.

1 Lab Overview

In this lab, you will learn about how AI/ML can be used to detect societal issues such as cyberbullying. Cyberbullying is bullying performed via electronic means such as mobile/cell phones or the Internet. The objective of this lab is for students to gain practical insights into online harassment, such as cyberbullying, and to learn how to develop AI/ML solutions to defend against this problem.

In this lab, students will be given a starter code. Their task is to follow the instructions provided in the Jupyter Notebook, train an AI/ML model on the given dataset, evaluate their model, and deploy the model by testing it on their own samples. In addition to the attacks, students will also be guided to perform hyperparameter tuning to improve the performance of their detection models further. Students will be asked to evaluate whether their tuning effort improves their detection models or not. This lab covers the following topics:

- Detection of cyberbullying in text.
- Hyperparameter tuning to affect model performance.

Disclaimer: This lab and AI models are intended for education and research purposes only. The lab could contain potentially triggering language and deal with difficult subject material. We have minimized showing such language samples in this lab. They do not represent the views of the authors.

2 Lab Environment

This ADVANCE lab has been designed as a Jupyter notebook. ADVANCE labs have been tested on the Google Colab platform. We suggest you use Google Colab, since it has nearly all software packages pre-installed, is free to use, and provides free GPUs. You can also download the Jupyter Notebook from the lab website, and run it on your own machine, in which case you will need to install the software packages yourself (you can find the list of packages on the ADVANCE website). However, most of the ADVANCE labs can be conducted on the cloud, and you can follow our instructions to create the lab environment on the cloud.

3 Lab Tasks

3.1 Getting Familiar with Jupyter Notebook

The main objective of this lab is to learn how AI/ML can be used to detect online harassment, such as cyberbullying. Before proceeding to that, let us get familiar with the Jupyter Notebook environment.

Jupyter notebooks have a Text area and a Code area. The Text area is where you'll find instructions and notes about the lab tasks. The Code area is where you'll write and run code. Packages are installed using `pip` and need to be preceded with a `!` symbol. Try accessing the lab environment for this task here.

The lab has three areas: one text area and two code areas. Follow the instructions for the three areas, fill the three areas with the instructed content, and add a screenshot to your report.

3.2 Cyberbullying Detection

In this lab, you will develop AI to detect cyberbullying. You will use a dataset of tweets to train your AI model, evaluate the performance of your AI model, and then deploy it by running it against your own samples. You can access the lab by clicking here.

3.2.1 Datasets Selection

In this lab, we provide three datasets: the Formspring dataset, the Davidson dataset, and the Founta dataset. You can edit the name of the dataset in the following code:

```
main_df = pd.read_csv('CyberbullyingLab1/formspring_dataset.csv', sep = '\t')
```

In this lab, you will be using the Formspring dataset, a widely used dataset of cyberbullying texts. Run all the code until this part of the lab. Report the size of the training, testing, and validation sets.

3.2.2 Preprocessing Data

Follow the instructions in the text areas and run the subsequent codes to preprocess your data, as follows. Here is a sample from the lab:

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
sentences = "the cat sat on the mat"
```

Add code to preprocess the following cyberbullying text, and include the generated tokens in your report: "`Harlem shake is just an excuse to go full retard for 30 seconds`". Please attach your code and show the output in your report.

3.2.3 Model Training

After you have preprocessed the dataset, the next task is to train the AI model. Follow the lab instructions to train the AI model. What is the final training accuracy that your model achieves? Please attach your code and show the output in your report.

3.2.4 Model Evaluation

Now it is time to run your trained model on a test dataset. Recall that we have already partitioned the dataset into train, validation and test sets. Run your model on the test partition and report your results here. Use the evaluate function.

```
..., ... = evaluate(...) # complete this code
print(f'| Test Loss: {test_loss:.3f} | Test Acc: {test_acc*100:.2f}% |')
```

3.2.5 Deploy and Run Custom Samples

Copy the sentences from the samples file and use your model to check if they contain cyberbullying content. Report the samples that were detected as cyberbullying. Do you think your model is good enough? In this lab, you will later learn how to use hyperparameters to improve the performance of your model.

3.3 Hyperparameter Tuning for Cyberbullying and Hate-speech Detection

In this task, some hyperparameters will be tuned to observe the effect on the model. Hyperparameters determine the parameters (weight and bias) of your AI model. Hyperparameters include learning rate α, number of epochs, number of hidden layers/dimensions, number of hidden units, choice of activation function, mini-batch size, etc. In this task, we will focus on learning rate and number of epochs.

3.3.1 Number of Epochs

Training AI models requires going through our dataset multiple times to learn the optimal parameters. Number of epochs is the number of times we go through our entire dataset in order to improve performance. However, an inappropriate setting may cause underfitting or overfitting of the AI models. In this task, we will observe how varying the number of epochs affects the model.

Keep the other hyperparameter as the default value we used. With setting `training_epochs` as 2 and 10, train the model. What are your observations? How did the changes affect the accuracy of the training and test datasets? After this experiment, what number do you think may be better for training epochs? Please show the plots in your report and describe your findings.

3.3.2 Learning Rate

Learning rate needs to be chosen carefully in order for gradient descent to work properly. How quickly we update the parameters of our models is determined by the learning rate. If we choose the learning rate to be too small, we may need a lot more iteration to converge to the optimal values. If we choose the learning rate to be too big, we may go past our optimal values. So, it is important to choose the learning rate carefully.

The given cell compares different learning rate values (`0.01`, `1e-3`, and `1e-5`). A model is trained for each of the learning rate values and the metrics stored. For each of the learning rates, the training loss is plotted against the number of epochs. Observe the outputs and report your observations.

3.3.3 Discussion

We experimented with different hyperparameters in this lab, what can you conclude about training AI models? Specifically, what are your observations about the model before Vs. after hyperparameter tuning?

4 Submission Instructions

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanations for the observations that are interesting or surprising. Please also list important code snippets followed by explanations. Simply attaching code without any explanation will not receive credits.