Homework 2: Abusive Language Detection

Due dates: part1 due March 17th, part2 due March 20th, both are due @ 2:00pm CDT

Although social media platforms have provided an equal opportunity for online users to share their knowledge and interact with each other in an unlimited space, they also expose users to almost unlimited harassment. The goal of this assignment is to build an RNN-based deep neural classifier that can identify aggressive messages in online posts.

Task

Your task is to implement an abusive language classifier using Recurrent Neural Networks (RNNs). Your code should be implemented in Python 3 and you are allowed to use available libraries such as pytorch.[1] The main modules you need to implement to achieve the goal of this assignment are: 1) **Data pre-processor** to transform raw data into a format suitable for input to the model, 2) **Token encoder** to represent each token with a vector of real numbers, 3) **RNN classifier** to assign a label to each input sequence that shows whether it is offensive or not, 4) **Training function** that performs forward and backward propagation, and 5) **Evaluation function** that evaluates the performance of the model in every training epoch.

Recommended readings about RNNs:

- Jurafsky & Martin: Chapter 9
- Illustrated Guide to LSTM's and GRU's: A step by step explanation
- Understanding LSTM Networks

Data

For this assignment, the data comes from Facebook. The format of the data is the same across all three different sets. Each row includes the document id, comment, and label. There are three different classes in this data:

- **Non-aggressive (NAG):** There is no aggression in the text.
  Example: no permanent foes, no permanent friends. interest is permanent!
- **Overtly aggressive (OAG):** The text is containing either aggressive lexical items or certain syntactic structures.
  Example: You can not escape from the sin of promoting this useless fellow.
- **Covertly aggressive (CAG):** The text is containing an indirect attack against that is not explicit, i.e., not using swear words or other direct forms of attack.
  Example: Absolutely! the deeper you dive the shallower cushion you have.

[1] https://pytorch.org

Note that you do not have access to the actual labels for the test set. Data can be downloaded via the following links:

- Training set
- Development set
- Test set (no label)

You should use training data for training your model, and development set for fine-tuning the hyper-parameters of the neural network. You can also use your development set for evaluating and analyzing your model. Test data must only be used for the final evaluation of your model.

Baseline

We ran a simple baseline to give you some idea regarding the performance of the model. The baseline uses unigram features with a Logistic Regression classifier. This model was trained on training set and evaluated on development data.

Here are the results:

```
     precision    recall  f1-score   support

CAG       0.51      0.50      0.51       700
NAG       0.67      0.65      0.66       815
OAG       0.50      0.54      0.57       485

avg/total 0.57      0.57      0.57      2000
============================================================
Confusion matrix
============================================================
['CAG' 'NAG' 'OAG']
[[352 182 166]
[188 532 95]
[144 80 261]]

Test Accuracy:  0.573
Macro Precision Score, 0.561548, Micro Precision Score, 0.572500, Weighted Precision Score, 0.574402
Macro Recall Score, 0.564587, Micro Recall Score, 0.572500, Weighted Recall Score, 0.572500
Macro F1-score, 0.562774, Micro F1-Score, 0.572500, Weighted F1-Score, 0.573211
Misclassified samples:  855
============================================================
```

Deliverables

This assignment is to be completed individually. You will need to turn in the following:

1. **Part1 (prediction files):** This part is due on **March 17th by 2:00 PM (CDT)**. You need to deliver your test predictions in a csv file: `test_prediction.csv`. You need to have two columns in this file: ID and Label. You can find a sample prediction file here. You should submit this part via BlackBoard. Every one can submit up to three different prediction files using your three best models. Note that the

number of submissions can be 1, but you have the option to submit up to 3. Please put all your prediction files (up to three) into a single zip file:
`lastname_COSC6336_hw2_part1.zip`
and include the csv file with the predictions and a readme file per csv file that briefly describes the specific features/model you have used for generating that specific output. For the submissions, please take into account the following requirements:

- Keep the order of the documents the same as the original test set.
- Use exactly the same format for your submissions as what we described above.
- Use the same **pseudonym** that you chose for HW1. Make sure that you add this name to your readme file(s). We will evaluate your models using Macro F1 metric. Then, we will compare the performance of your best submitted model to other students' and rank your system based on that. The final ranking will be posted to Piazza on **March 18th by 2:00 PM**, and will be anonymized using the pseudonym you have provided.

NOTE: Make sure you follow the format specifications; if the evaluation script breaks with your input you will loose any points associated with this aspect.

2. **Part2 (code & report):** This part is due on **March 20th, 2:00 PM (CDT)**, and you need to deliver the following items:

   - **Code:** should be written in python 3.
   - **Report:** you have to provide a technical report in pdf format describing your system, relevant experiments, and analyzing the results. The report should be written in ACL style format. You can either download the template here or open it on Overleaf. **Do not include code on your report**.

   You need to submit this part as a zip file named `lastname_COSC6336_hw2_part2.zip` to BlackBoard.

Grading Policy

For this assignment, we will use the following grading percentages:

- Overall performance of your model: 10%
- Originality of the model: 30%
- Analysis of results and findings: 30%
- Overall quality of write-up: 30%

If you have any questions regarding the assignment, please post it on Piazza or see us during office hours.