Multiclass Noisy Text Classification

In this assignment, you will develop a multiclass classifier for predicting the most probable hashtags for given tweets, considering the noisy and imbalanced nature of social media text. We consider a supervised learning setting where a small dataset of training and validation tweets and their corresponding hashtags is provided. Note that hashtags are user-generated content and may not be precise or comprehensive. However, hashtags of *test* tweets are precise and comprehensive as they are verified by human annotators in the hashtag space of this assignment. The final goal is to achieve high *balanced accuracy* in hashtag prediction. The dataset is available on Blackboard:

- `hashtags.txt`: List of possible hashtags.
- `train.txt`: Training tweets with de-identified IDs, followed by their hashtags and content.
- `val.txt`: Validation data, formatted like `train.txt`.
- `test.txt`: Test data, similar to `train.txt` but without hashtag IDs.

Note that the dataset is imbalanced across different hashtag classes, data splits are stratified based on hashtag frequencies, and test tweets are matched with one most probable hashtag for simplicity.

Performance Improvement Tips Human language is complex. For example, although the tweet "tested flipping a coin using null hypothesis" contains the word *coin*, it only matches the hashtag `#datascience` but not `#cryptocurrency`, or the tweet "I'm using Gate.io to buy and sell bitcoins" is mainly concerned with `#cryptocurrency` but not `#datascience`. Your classifier should perform well on these harder examples to obtain an overall good performance. Try to identify and implement a set of informative, discriminative and generalizable features. Some good ideas to try are: (a): use *Twitter Word Clusters*[1] to partially account for the various non-standard ways that the same word is written, e.g., the following words have often been used to refer to the word *know* on Twitter: {*knoww, knowww, knowwww, knooow, knoow, knoooow, knowwwww, kow, knwo, knoooooow, knowwwwww, konw, knooooooow, knooww, knowwwwwww, kniw, kmow, knooowww, knnow, lnow, knowwwwwwwww, knoowww*} but they are considered as different words/features by most machine learning algorithms; (b): learn a policy, e.g. using Contextual Bandit in `vowpal wabbit`, to estimate the complexity of samples and devise a "curriculum" to better train the classifier, e.g., start the training with easier samples and then gradually proceed with more difficult ones. This training paradigm is inspired by a process (called *shaping*) by which humans and animals acquire knowledge and skills and is often referred to as *curriculum learning*.[2]. You can try this idea using `vowpal wabbit` as it allows loading a pre-trained model and continuing training with new data. Ensure reproducibility by fixing your random seed to 123.

Assessment Criteria The assessment will focus on the depth of the analyses conducted, with specific attention to the following aspects:

- **Code Quality and Efficiency (10%):** Organization, readability, and computational efficiency.
- **Features and Model Innovation (50%):** The creativity and rationale behind the choice of features extracted from the tweets, and the selection or design of the model architecture for classification. Special attention will be given to how these choices improve the classifier's performance on noisy text in tweets.

[1] `http://www.cs.cmu.edu/~ark/TweetNLP/cluster_viewer.html`

[2] See this paper for more information: Bengio, Y., Louradour, J., Collobert, R. and Weston, J., 2009, June. Curriculum learning. ICML'09. `https://ronan.collobert.com/pub/2009_curriculum_icml.pdf`.

- **Handling Imbalanced Data (10%):** Strategies implemented to mitigate challenges of learning with imbalanced data, including but not limited to resampling techniques or innovative loss functions.
- **Results and Analysis (30%):** Models will be compared based on *balanced* accuracy[3] in predicting the hashtags of test tweets. Provide insights on performance trends observed in relation to specific hashtags, potential areas of improvement, and describe any findings.

Submission Instructions

Submit a zip file named `[STUDENTID].zip` with the following files in root directory:

1. `classifier.py`: Your classification script. Submissions must be compatible with Python3 and should not be in Jupyter Notebook format. Convert notebooks using:
   ```
   $ jupyter nbconvert --to script [NOTEBOOK].ipynb.
   ```
2. `predictions.txt`: Output file with the predicted hashtag ID for each test tweet; each line should contain the test tweet id followed by TAB followed the most probable hashtag class id for the tweet.
3. `README.txt`: brief explanation of the approach and simple instructions to run the classifier. We expect an EASY-to-run script. Specifically, in `Python`, we will run the following command, which should generate `predictions.txt` in the above-mentioned format:
   ```
   $ python3 classifier.py -d train.txt -v val.txt -t test.txt
   ```

Good luck with the assignment!

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html`