HW03 – Language Models

**Observations Criteria:**

Note:

* The code is aimed at training language models using the Maximum Likelihood Estimation (MLE) algorithm and testing these models on different test datasets.
* The code loads data from CSV files, preprocesses it by converting it to lowercase and removing newline characters, trains an MLE model on the preprocessed training data, and tests the trained model on the preprocessed test data.
* The training data and test data are loaded from separate CSV files, and the test data is divided into three separate datasets: test\_full, test\_not, and test\_toxic.
* Three language models are trained: LM\_full on the entire training dataset, LM\_not on only the non-toxic comments in the training dataset, and LM\_toxic on only the toxic comments in the training dataset.
* The test\_LM function is called to test each of these language models on each of the test datasets.
* Overall, the code is well-organized and uses the MLE algorithm provided by the nltk.lm library to train language models.
* The code also provides a method for testing the language models by calculating the logscore of bigrams in the test dataset.
* The code outputs the results of the tests to CSV files, making it easy to analyze and compare the performance of the different language models.

CSV File (converting it to token and lowercase)

Date loaded from separate datasets & LM models trained

Test\_LM function called

MLE algorithm called by nltk library

Result of the output flashed in new csv file

**Summary:**

* This code is an implementation of language models for classifying toxic comments in a dataset. Took many help from different open source website like youtube, stackoverflow, geekforgeeks etc.
* The code loads train and test data from CSV files and preprocesses the text data by converting to lowercase and removing newline characters.
* It then uses the NLTK library to train a language model using maximum likelihood estimation (MLE) on bigrams of the training data.
* The code also defines a test function that takes the trained language model and test data, and outputs a score for each comment. The output scores are then written to a CSV file.
* The code creates three different language models based on different subsets of the training data: one for all comments, one for non-toxic comments, and one for toxic comments.
* The test function is then called on each of these language models using the test data, and the results are written to separate CSV files.
* However, got an unexpected error from the code when doing the test from train\_LM. Also, there is a minor encoding error happened in the last. So, could not get the perfect result from this code
* Overall, this code is a simple implementation of language models for text classification and can be useful as a starting point for more advanced models.

Thank you!

Sincerely,

Rifat Jahan Rezwan