Foundations of Natural Language Processing
Peking University, 2023

Homework 2: Due Sunday, April 7, 2023 by 11:59 pm

PLEASE read these instructions to ensure you receive full credit on your homework. Submit your homework as a zip file through **Course**, which should include one report in PDF, your source code in python, and one shell script. The report should be generated from the LaTeX template in the attachment and you do not need to submit the data you used. The shell script will be used to start your program and a detailed description is at the end of this document. Your grade will be based on the contents of one PDF file, the original source code, and the script. Additional files will be ignored.

LATE SUBMISSION POLICY Late homeworks will have 5% deducted from the final grade for each day late. No submission will be accepted after **April 14**, one week after the due date. Your homework submission time will be based on the time of your **last** submission to Course. Therefore, do not re-submit after midnight on the due date unless you are confident the new submission is significantly better to overcompensate for the points lost. You can resubmit as much as you like, but each time you resubmit be sure to upload **all** files you want to be graded! Submission time is non-negotiable and will be based on the time you submitted your last file to Course. The number of points deducted will be rounded to the nearest integer.

Problem Description

In this homework, you will implement a log-linear model for a text classification problem. Given a document **d**, you should first extract some features from it, like uni-grams and bi-grams in **d**. You can also extract other features. After that, to classify the document into an appropriate group, you will build a log-linear model from scratch and implement the update algorithm by yourself, **You are prohibited to use any toolkit to implement the log-linear model. No sklearn, pytorch, tensorflow or other package similar to them. Your score will be zero if we find you using such package.** Accuracy and Macro-F1 can be used as the measurements.

We choose *20-Newsgroups* as our dataset, which is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The dataset is split into train and test set, stored in train.csv and test.csv. The first column in the data file is document text and the second column is the corresponding group label. You can find label names in label.csv.

The codes should include the following parts:

- Data preprocessing.

- Feature extraction.
- The implementation of log-linear model.
- The update algorithm.
- Evaluation. (**The evaluation function should be implemented by yourself. To check the correctness of your implemention, compare it with sklearn.metrics.f1_score and sklearn.metrics.accuracy_score. Please make sure that the results from your implementation match the results from sklearn. In the report, please include the results of these two types of evaluations.**)

The experiment report should include the following parts at least:

- The implementation of log-linear model.
- The update algorithm.
- The results on both train and test set.

Description of Shell Script

To make sure we can run your code successfully, you need to submit a shell script, named ***run.sh***. We will run this script in the terminal to start your program. If we cannot start your program from the script, you can only obtain 50% of the total score at most. You can assume this script, your source code and data files are in the same directory. Please use **Relative Path** rather than **Absolute Path** in both your script and your source code.