EECS595: Natural Language Processing
Homework 1 Programming Assignment

Assigned date: 9/3/2021 ==== Due date: 9/17/2021

September 3, 2021

The goal of this programming assignment is to get you warmed up with Python programming. You can use packages such as `numpy`, `re`, and `nltk`. This can be an opportunity for you to learn how to use common tools for NLP systems.

1. In a textual document, some textual strings may have special semantic meanings which relate to different named entities such as person names, location names, organization names, times, dates, email addresses, web addresses, dollar amounts, etc. We have seen in the class that regular expressions can be used to match a text string to identify these named entities related to that text string. For example: /$[0-9]+˙[0-9][0-9]/ can be used to match any string indicating some dollar amount, e.g., "$149.99". Once a match is found, the system can assign a special name (e.g., DOLLAR_AMOUNT) to the string (i.e., "$149.99").

   Write a Python program where you will specify regular expressions to identify the following types of named entities: TIME, DATE, EMAIL_ADDRESS, WEB_ADDRESS, DOLLAR_AMOUNT. Define a function called `ner(input)` which will output the sentence that replaces the special strings in `input` to their recognized named entities.

   For example, if `input` is:

   > "She spent $149.99 on a nice microphone from `www.bestdevices.com` yesterday."

   `ner(input)` should return:

   > 'She spent DOLLAR_AMOUNT on a nice microphone from WEB_ADDRESS DATE."

   We will not give you any training data. You should use your knowledge to come up with the appropriate regular expressions. Try to cover as many possibilities as you can think of. Your program will be tested on a small set of testing examples.

   For your convenience, a skeleton Python program is provided for you: `namedentity_skeleton.py`. You may use it as needed.

   Submit through CANVAS:

   - Your python program named `namedentity.py`.
   - A readme file `README_1.pdf` answering the following question: what are the potential problems with this approach to named entity recognition?

2. The second assignment is to implement a simple naïve Bayes classifier for sentiment analysis. We will use a standard movie dataset, provided on Canvas in `HW1_programming_data.zip`. In the `training` and `testing` directories, there are two subdirectories `pos` (which consists of documents labeled as having positive sentiment) and `neg` (which consists of documents with negative sentiment).

   You will use the `training` data to learn model parameters using features of your choice (e.g., simplest features are bag of words). Your trained model will be applied to the `testing` data to predict the label for each testing document. Ground-truth labels are also provided in the testing data, so you should use them to calculate the performance of your model using *accuracy*.

   Your program should consist of three modules:

   (a) A *training module* that learns the parameters

   (b) A *testing module* that applies the learned model to make predictions on the testing data

   (c) An *evaluation module* that calculates the accuracy of the predictions made by the model

   In this assignment, you have the freedom to use any pre-processing you see fit and any features of your choice. The README file provided with the dataset provides some background on the kind of features you may consider. Explain the set of features you have applied and report your results in your submitted README file.

   For your convenience, a skeleton Python program is provided for you: `naivebayes_skeleton.py`. You may use it as needed.

   Submit through CANVAS:

   - Your Python program named `naivebayes.py`
   - A readme file `README_2.pdf` which explains your features and reports your results.