COMP 598 Homework 3 – MLP Conversation Analysis

30 pts
Assigned Sept 24, 2020
Due Oct 2, 2020 @ 11:59 PM

This is an INDIVIDUAL Assignment – each student's work must be their own, each student completes this assignment, there are no teams for homework 3.

The goal of this assignment is for you to develop python scripts and code using best practices covered in the lessons this week to conduct a complete a data analysis project on My Little Pony. Note that all work for this homework must be done in python.

Task 1: Watch some My Little Pony episodes (0 pts – totally optional)

It's always important to study your source material … particularly when it's very entertaining cartoons!

Task 2: My Little Pony dialog analysis (20 pts)

We'll be using the dataset available here: https://www.kaggle.com/liury123/my-little-pony-transcript

For the purpose of this study, we'll use only clean_dialog.csv and assume that the dataset is perfect.

Write a python script named analysis.py that, when run, computes and produces a JSON-formatted analysis of the ponies' interpersonal dynamics that has exactly the structure given below (all numbers below are just examples). The canonical pony names used in the file should be: twilight (Twilight Sparkle), applejack (Applejack), rarity (Rarity), pinky (Pinky Pie), rainbow (Rainbow Dash), and fluttershy (Fluttershy). All other characters are considered "non-Pony" characters.

```
{
  "verbosity": { // give fraction of dialogue, measured in # of speech acts produced by this pony
    "twilight": 0.37,
    "applejack": 0.24,
    ...
  },
  "mentions": { // give fraction of times each pony mentions the other
    "twilight": { // the fractions here should sum to 1
      "applejack": 0.12,
      "pinky": 0.51,
      ...
    },
    ...
  },
  "follow_on_comments": { // the fraction of times each pony has a line that DIRECTLY follows the others pony's line
    "twilight": { // the fractions here should sum to 1
      "applejack": 0.21,
      ...,
      "other": 0.4 // this is the number of times TS has dialogue following a non-Pony character
    },
    ...
  }
  "non_dictionary_words": { // a list of the 5 non-dictionary words used most often by each Pony
    "twilight": [ "huh", "ugh", "awwww" , "wheee", "wha"]
    ...
  }
}
```

e.g. all sentences spoken by pinky should be combined into one entry (speech act)

this is just a fake e.g.

300 192 150

should be ordered

Attend to the following details:

- Here a "word" is any substring bordered by non-alphanumeric characters OR the start/end of the containing string. This means that "anti-aircraft" contains the words "anti" and "aircraft".
- A pony mention occurs when any of the words composing that pony's name appears in dialog, with that word capitalized. So "Hey Twilight!" counts as a mention of Twilight Sparkle. "I like pie" does not count as a mention of Pinky Pie because "pie" is not capitalized.
- Non-dictionary words are any not present in the list words_alpha.txt, located here: https://github.com/dwyl/english-words
    - This should be saved in your project as data/words_alpha.txt

specify address when doing assignment but DONT submit this!!!

Task 3: Unit Testing (5 pts)

Write at least 10 unit test (10 functions) for your code spread across mentions, follow-on-comments, and non-dictionary words. They must all pass.

Note on grading for unit tests: the TAs will spot check your tests to confirm that they aren't just a trivial self.assertTrue(True). Beyond that, it's up to you to think about what to test and how to test it – we won't be checking this deeply. I encourage you to compare unit tests with other classmates or come to office hours to discuss.

Submission Instructions

Your MyCourses submission should contain a project with the following structure

- scripts/
    - analysis.py
        - This should use argparse and print a helper message when no arguments are given.
        - This should accept the link to the clean_dialog.csv.
        - It should assume that words_alpha.txt is sitting in the data/ directory.
        - It will be run in a UNIX shell in which PYTHONPATH includes a path to the project's src directory. This will allow it to use code in the hw2 package.
        - It should accept an optional argument "-o <file_name>". If given, the JSON output is written to that file. If it is NOT given, the JSON output should be written to stdout.
- data/ - this directory is empty. Do NOT submit your dialogue or words files. When graded, the TAs will provide these.

Just print out???

    - Nothing in this directory.
- src/
    - hw2/
        - <code>
        - test.py – this runs all your unit tests. At least 10 must be run and succeed.
        - tests/ - this directory contains your unit tests