COMP 598 Homework 9 – Using TF-IDF

30 pts
Assigned Nov 19, 2020
Due Nov 27, 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 9.

Non-standard (i.e., built-in) python libraries you can use:

- pandas
- requests

In this assignment, we're going back to homework 3 and computing the each pony's most frequent words using TF-IDF. Note that, throughout this assignment, we refer to "pony names" – use the canonical names we used for each of the main character ponies in HW3.

Task 1: Compute word counts (15 pts)

Write a script that computed word counts for each pony from all episodes of MLP. Your script, compile_word_counts.py should run as follows:

python3 compile_word_counts.py -o ../data/pony_counts.json clean_dialog.csv

python compile_word_counts.py -o <word_counts_json> <clean_dialog.csv file>

The output file should be a dictionary with the following form:

```
{
    "<pony-name>": {
        "<word>": <# of times it appears>,
        "<word>": <# of times it appears>,
        …
    },
    "<pony-name>": {
        …
    }
}
```

Note that in the output file you should ONLY have word counts for words that are spoken by the pony at least five times. Toss out any words that happen less than this many times (this is a common technique to avoid storing information on a crazy number of words).

Other details:

- Only consider speech acts involving one of the main character ponies. Ignore any others.
- Treat each word encountered as case insensitive. Store words in all lowercase form.
- Before processing text, replace punctuation characters with a space – a punctuation character is one of these: ( ) [ ] , - . ? ! : ; # &
- A word must only include alpha-only characters. All other words should be ignored.
- Tip: to keep your script performant, store your word counts in dictionaries.

Task 2: Compute most frequent & distinctive pony language (15 pts)

Write the script compute_pony_lang.py which is run as follows:

python compute_pony_lang.py <pony_counts.json> <num_words>

python3 compute_pony_lang.py pony_counts.json 2

python3 compute_pony_lang.py -p pony_counts.json 2

The <pony_counts.json> file should have the same format output by your compile_word_counts.py script. It should compute the <num_words> for each pony that have the highest TF-IDF score, using the definition of TF-IDF given in the lecture.

Output should be written in JSON format to stdout with the following structure:

```
{
    "<pony name>": [ "highest-tfidf-word", "second-highest-tfidf-word", … ],
    "<pony name>": …
}
```

Each pony word list should have <num_words> entries.

Submission Instructions

Your MyCourses submission must be a single zip file entiled HW9_<studentid>.zip. It should contain the following items:

- scripts/
  - compile_word_counts.py – script for Task 1
  - compute_pony_lang.py – script for Task 2