Project 1: Tweet Mining & The Golden Globes

Project Deliverables:

1. All code must be in Python 3. You can use any Python package or NLP toolkit, but please save and share your requirements as follows: Create an environment for the project, run "pip freeze > requirements.txt", make sure it works after running "python install -r requirements.txt" in an empty environment, and then include this "requirements.txt" in your submission/repository.
2. Please use the Python standard for imports described here: https://www.python.org/dev/peps/pep-0008/#imports (Links to an external site.)
3. Bundle all your code together, your submission will be a .zip file.
4. Your code must be runnable by the TA: Include a readme.txt file with instructions on what file(s) to run, what packages to download / where to find them, how to install them, etc and any other necessary information.
5. Your code must run in a reasonable amount of time.
6. Your code cannot rely on a single Twitter user for correct answers. Particularly, the official Golden Globes account.

Minimum Requirements:

A project must do a reasonable job identifying each of these componentes:

1. Host(s) (for the entire ceremony)
2. Award Names (Categories)
3. Presenters, mapped to awards*
4. Nominees, mapped to awards*
5. Winners, mapped to awards*

*These will default to using a hardcoded list of the awards to avoid penalizing you for cascading error. Please note that, when mining award names specifically, you cannot hardcode parts of these names in your solution with the only exception of the word "Best."

It is OK not to have 100% accuracy. It's very rare for any group not to have some error, especially with awards and nominees. Even getting just half of the nominees for a given award is quite good performance.

Additional Goals:

Some examples of additional goals:

- Red carpet – For example, determine who was best dressed, worst dressed, most discussed, most controversial, or perhaps find pictures of the best and worst dressed, etc.
- Humor – For example, what were the best jokes of the night, and who told them?

- Parties – For example, what parties were people talking about the most? Were people saying good things, or bad things?
- Sentiment – What were the most common sentiments used with respect to the winners, hosts, presenters, performances, and/or nominees?
- Performances – What were the performances, who were the performers, when did they happen, and/or what did people have to say about them?
- Your choice – If you have a cool idea, suggest it to the TA! Ideas that will require the application of NLP and semantic information are more likely to be approved.

Required Output Format:

You are required to output your results in two different formats.

1. A human-readable format. This is where your additional goals output happens. For example:
   Host: Seth Meyers

   Award: Best Motion Picture - Drama
   Presenters: Barbara Streisand
   Nominees: “Three Billboards Outside Ebbing, Missouri”, "Call Me by Your Name", "Dunkirk", "The Post", "The Shape of Water"
   Winner: “Three Billboards Outside Ebbing, Missouri”

   Best Dressed: Jane Doe
   Worst Dressed: John Doe
   Most Controversially Dressed: John Smith

2. A JSON format compatible with the autograder; this is only containing the information for the minimum tasks. For example:

{

"Host" : "Seth Meyers",

"Best Motion Picture - Drama" : {

"Presenters" : ["Barbra Streisand"],

"Nominees" : ["Three Billboards Outside Ebbing, Missouri", "Call Me by Your Name", "Dunkirk", "The Post", "The Shape of Water"],

"Winner" : "Three Billboards Outside Ebbing, Missouri"

},

```
"Best Motion Picture - Musical or Comedy" : {

"Presenters" : ["Alicia Vikander", "Michael Keaton"],

"Nominees" : ["Lady Bird", "The Disaster Artist", "Get Out", "The Greatest Showman", "I, Tonya"],

"Winner" : "Lady Bird"

},
```

The Data:

gg2013.json.zip

You will be graded on at least one year you have not seen.

The Autograder:

The autograder is your way of benchmarking your progress as you work on improving accuracy.

- The central repository is at https://github.com/c-col/gg-project, and it contains:
    - A copy of the autograder program, which will assess how well you did on the basic tasks.
    - A template for the API the autograder uses, saved as gg_api.py. Be sure to read the doc strings and ask the TA if you have any questions about how to use this file.
    - A JSON file with the correct answers for the minimum task for 2013; these answers are used by the autograder. DO NOT read this into memory in your own code. **Doing so is grounds for an automatic zero.**

The autograder is somewhat strict. Use it to guide your development. When we actually grade your project, we'll look "near misses" as well as how well you've considered generality in architecting your system.

Grading:

1. Your code will be run and graded on at least one year for which you have not been provided data. This is to ensure you don't overfit to the provided data.