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. You must use a publicly accessible repository such as Github, and commit code regularly.
3. Bundle all your code together, your submission will be a .zip file.
4. If you use a DB, it must be Mongo DB, and you must provide the code you used to populate your database.
5. 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. The readme should also include the address for your Github repository.
6. Your code must run in a reasonable amount of time.
7. 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 components.

1. Host(s) (for the entire ceremony)
2. Award Names
3. Award categories; Presenters
4. Nominees, Favorites, Winners 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.

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:

- Excerpts from Winners' statements; reactions
- Other honors, speeches
- Unexpected events
- Visuals (i.e., build a slide show)?

Note: Your system should be easily adaptable via configuration or setting of meta-parameters to other awards ceremonies, specifically, the Academy Awards.

The Data:

Uploaded to CollPoll

The Autograder:

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

> The master repository is at https://github.com/milara/gg-project-master (Links to an external site.), and it contains:
>
> - A copy of the autograder program, which will assess how well you did on the basic tasks. It has undergone some changes as the project format has changed, so please report bugs early and often so that I can get it fixed ASAP.
> - 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.
> - JSON files with the correct answers for the minimum tasks for both 2013 and 2015; these are used by the autograder. DO NOT read this into memory in your own code. **Doing so is grounds for an automatic zero.**