Final Project

- Write your own NLP Application
    - Unlike with the exercises it needs to do more than off-the-shelve code
    - Needs to have many or all of the following:
        - A web front end
        - A database back end
        - Unit tests
        - Documentation

Final Project

- Use Python
  - But you may wrap non-Python code
- It should be easy for a user to install and run your application
  - Typically means using Docker or creating a package
- Work in groups of 2-4
- First plan out your work

Tentative Timeline

March 22	Send me a few paragraphs on your group and what you want to do.
March 26	Have a 2-4 page plan. Be ready present this in class. (this may be too ambitious)
April 16 & 19	10-15 minute presentations in class
May 5th	Deadline No extensions possible for those who graduate in May

Write up

- Report
  - what you did and why
  - how you implemented it, broadly
  - what went right and what went wrong
  - who implemented what
- README.md file
  - how to run

Check list

- On the NLP component:
  - What are you trying to accomplish?
  - How is it different from previous approaches?
  - Can we analyze the differences?
  - How do you measure how good it is?

Check list

- From a coding/testing perspective:
  - What parts of the code can I meaningfully test?
  - How do I use unit tests (or other tests) for that?
  - How do we as group members divide the work?
  - What workflow do we have for our GitHub repository?

Check list

- From the user’s perspective:
  - Who is our user?
  - What experience do we want the user to have?
  - How can we wrap our application so that it is easy to use?
  - Note:
    - No need to go overboard here on the first two items, it is okay to say that the user has some sophistication in using computers. Think of yourself as you were before you took this class.

What to deliver? A GitHub Repository.

- NOT the one you used for the assignments
- If it is private, make sure Yangyang (Brendayy) and I (marcverhagen) have access
- It should contain your report and everything needed to run your application
- Please think about how you structure your repository
    - Docs, code, readme, license
- Send email when you are done

Examples from yesteryear

- Natural language interface to relational database
- Haiku bot
- Twitter classification
- Auto Badminton Brief Report Generator
- Text normalization (currencies)
- Detecting challenging or uncommon words
- Question-answering system
- Spanish chat box
- Movie review classification
- Generating rhyme patterns