LING 331 - Text Processing for Linguists

Week 1

Intro,
Unix, Shell,
Environment, Files
Who are we?

Rob Voigt
robvoigt@northwestern.edu
Assistant Professor
of Linguistics
and Computer Science
(by courtesy)

Grace LeFevre
GraceLeFevre2026@u.northwestern.edu
PhD Student
in Linguistics
Who is this class for?

- Linguists, social scientists, humanists
- Little-to-no programming experience
- Applications to research

Goals

- Lots of hands-on practice
- Teach you how to teach yourself
Who is this class not for?

- Folks with lots of programming experience
- CS Majors (probably - email me if this is you)
- COMP_SCI 110 is similar in focus (and uses one of the same textbooks) - what’s different?
  - CS110 - broad, more CS-y (e.g. debugging and testing)
  - LING331 - narrow focus on applications to text, we will purposefully skip less-relevant stuff
What will we learn?

● Unix Command Line
  basic usage, remote access, and tools for text

● Basic Python
  programming concepts, syntax, useful libraries for text

● Applications (as much as we have time)
  web scraping, APIs, data munging, text analysis
When and where will we see each other?

Here! Annenberg G30, TTh 11:00am-12:20pm.
short lectures, small group workshopping

Office hours  
Rob  
Tuesdays 1-2pm and by appt

Grace  
Thurs 10-11am and Fri 11am-noon

Ed discussion board for questions
help each other out!
COVID considerations

I have little kids - childcare loss is possible and/or likely

I will prioritize not missing class, but be aware it could happen

This is not a hybrid class - we will do a lot in person!

However, all classes will be recorded on Zoom on Canvas
Why are we doing this?

1. Get computationally “free” - GUIs only let you do things someone else decided on

2. Processing text data is useful for anyone’s research/work

3. This is the start of computational linguistics! data science, web search, speech-to-text, conversational AI, “big data” language analysis, etc etc
How will we do it?

Syllabus on course website:

https://faculty.wcas.northwestern.edu/robvoigt/courses/2023_winter/ling331/

Assignments, peer review, final project

Videos/readings before class;
working on assignments during class in small groups
How will we do grading?

Heavy emphasis on qualitative feedback:
   Grace and/or I will read your work and comments
   and provide qualitative feedback inline.
   No comment = “good job!”

Letter grades ultimately based on effortful completion,
   Midterm and final self-evaluations

The point of this whole thing is for you to learn, period!
What constitutes strong performance?

There is a lower bound:
  Do basic reading/watching of course material
  Complete basic assignment (make it work)

There is no upper bound:
  Each week will have extra material listed for reference
  Assignments will often have a number of possible extensions
  You can start working early on your final assignment
  Plus whatever you can dream up
Agreements

I see this class as entering into a set of mutual agreements, on top of the basic agreements of the university (academic honesty etc)

We’re building a community of learners interested in this topic! (I’m a learner too.)

By registering, you agree to certain things -
By being the instructor, I agree to certain things.
You agree to:

Invest substantial time and effort in this course this quarter
Hold yourself accountable for your own progress
Be honest in assignments, self-evaluations
Stay on top of your work, and ask for help when needed
Be open to constructive feedback
Challenge yourself
Communicate with me when any of the above falls through
I agree to:

Invest substantial time and effort in your process of learning
Prepare well for class, construct meaningful assignments
Make myself available to help
Be open to criticism and commentary
Provide structures for learning
Communicate with you when any of the above falls through
The Struggle!

Learning programming is like learning a new language

You have to soak in it and use it daily

It will feel unnatural at first, push through

Don’t be afraid to play around and break stuff
The Struggle Illustrated
YOU CAN DO IT

ERRORS ARE YOUR NEW FRIENDS

No such thing as a dumb question here.
Our new home: the command line
Precision - the challenge of exactitude

One wrong letter, space, or punctuation mark can easily derail you

These mistakes are at first very hard to see

Double-check, triple-check your code and relevant documentation
(a beloved acronym by programmers is RTFM - read the flippin’ manual!)

Take a break and come back to it
Benefits of command line interfaces

**Automatable**
- easy to do
- something 1000x

**Fast**
- GUI interfaces are computationally ‘heavy’

**Consistent**
- same command always does the same thing

**Transparent**
- you’ll learn what your files actually are
What is a file?

An abstraction!

... but ultimately, an array of bytes

e.g., for ASCII text:

<table>
<thead>
<tr>
<th>Character</th>
<th>L</th>
<th>I</th>
<th>N</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits</td>
<td>100</td>
<td>1100</td>
<td>100</td>
<td>1001</td>
</tr>
</tbody>
</table>
Types of Files

Text

bytes representing characters
txt, code (like .py), html, logs

Executable

compiled code in binary format
to run as a program

Data

everything else: images, zip files,
doc/ppt/pdf, and so on
Quest!

Remote computing environment, cluster of computers running Linux

Common for “big data” and high-performance tasks

Can schedule complex stuff, not waste your own machine

 scp assignment1.txt [netid]@quest.it.northwestern.edu:/projects/e31086/user/[netid]/assignment1/

Ideal to use Quest exclusively if you can
If it is slow for you at home, you can do everything locally, then upload assignments