Introduction to Computational Linguistics
Section

Olga Zamaraeva

University of Washington

April 3, 2020

Plan for today

- ▶ My notes on logistics etc
- ▶ Technical landscape overview

Prerequisites

- Basic linguistic theory (LING200)
  - Easy to "loose track" of it
  - Important for write ups
- Basic programming and probability skills are a big plus
  - But entirely possible for beginners to succeed
- Maturity when it comes to dealing with software

Prerequisites

- ▶ Basic linguistic theory (LING200)
    - ▶ Easy to "loose track" of it
    - ▶ Important for write ups
- ▶ Basic programming and probability skills are a big plus
    - ▶ But entirely possible for beginners to succeed
- ▶ Maturity when it comes to dealing with software
- ▶ The real prerequisite is **patience**
    - ▶ Curb your frustration :)
    - ▶ **Please** don't drop because you can't figure out command line

Reading highly technical material (Skip!)

- ▶ Identify the main idea/topic of the section/paragraph
- ▶ Try to relate this idea/topic to the bigger picture: Why is it important? How/where is it used?
- ▶ Articulate questions as you are reading
  - ▶ Go ahead and post them if you like!
- ▶ Reread some of the more confusing sections several times (it **does** help)
- ▶ Do not get bogged down; you do not have to understand every single word every single time

Course Logistics: Homework

- ▶ Some parts will be easy, some will be hard (for all backgrounds)
- ▶ Write ups are worth 50%; **please take them seriously**
- ▶ In your write up, show what you learned while completing the assignment; discuss challenges
- ▶ If you haven't completed the assignment fully, the write up is your chance to earn maximum **partial credit**
- ▶ If you did complete it, you still must submit a coherent and thoughtful write up in order to earn full credit.
- ▶ Jupyter notebook write ups should be as carefully written as if they were separate documents

Course Logistics: Coding

- ▶ Mostly python3
- ▶ You can use any editor, but IDE + debugger is highly recommended
- ▶ PyCharm community edition (free download)
  - ▶ https://www.jetbrains.com/pycharm/download/
- ▶ There are many (presumably good) tutorials on basic programming in python
- ▶ Jupyter notebooks (see instructions on the class site)
  - ▶ Optional to use; intended for beginners
  - ▶ If not helpful, do not waste time
  - ▶ (though do give the installation a try and post questions if you experience issues)

Computational Linguistics & NLP

CompLing == NLP?   CompLing ⊃ NLP?   CompLing ∩ NLP = ∅?
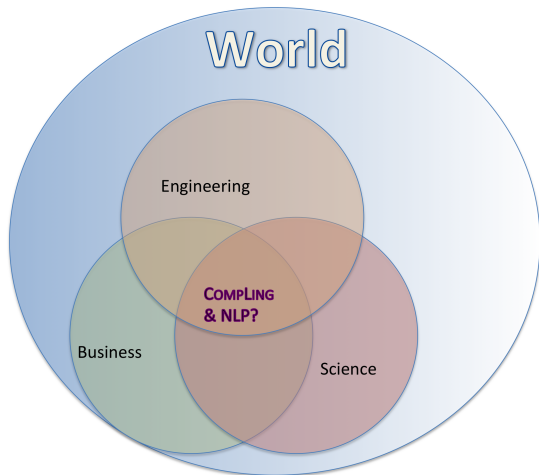
LING472

Section

Plan

My notes on the course

Logistics

CompLing and NLP

Technical overview

Assignment 0

Assignment 1

CompLing and NLP

- ▶ Computational Linguistics & Natural Language Processing
  - ▶ Analyzing language material to solve tasks and answer questions

- ▶ Natural Language Processing (NLP)
  - ▶ Focus: high resource data; algorithms
  - ▶ (Also: "Learning something about the world through data"?)
    - ▶ Classic objective: movie reviews → program labeling texts as having positive/negative sentiment
    - ▶ NLP researchers often refer to this as "computational linguistics"

- ▶ Computational Linguistics
  - ▶ Focus: study of language with computational means
    - ▶ Classic objective: descriptive grammar (book) → program behaving like in the book
    - ▶ Linguists often refer to this as NLP :)

Modeling language

"In any mathematical theory about an empirical domain, the phenomena of interest are modelled as mathematical structures, certain aspects of which are conventionally understood as corresponding to observables in the domain"

Pollard & Sag 1994

▶ A model is a formal (partial) description of an object/phenomenon

▶ What does it mean to model language?

▶ How close is a model to the real thing?

LING472

Section

Plan

My notes on the course

Logistics

CompLing and NLP

Technical overview

Assignment 0

Assignment 1

Technical demo

(Please refer to the class website for instructions)

- operating systems
- python
  - programming language
  - needed for almost everything
- command line
  - way of navigating your files and running programs
  - no GUI; ideal for some, annoying for others
  - needed for at least some assignments
- python IDE
  - Interactive Development Environment; a GUI for programming
  - to avoid command line at least in some portions and to debug efficiently
  - highly recommended
- jupyter
  - Interactive environment for running python code in chunks
  - experiment; supposed to help beginners

Getting help with coding

- Canvas (**please** try not to be shy)
- Google (usually leading to StackOverflow)
    - "How do I do X in python3?"
    - "I am getting error message Y"
    - Look at the most upvoted solution(s)
    - Requires you to build up some expertise but still helpful for beginners
    - What about **plagiarism**?
        - Can I use a line of code I found on StackOverflow?
        - Can I copy a function I found there?

Assignment 0

- ▶ Work in terminal (command line)
- ▶ Google the commands, e.g. "How to rename file in terminal"
- ▶ cd - change directory
- ▶ mkdir - make directory
- ▶ mv - move file (= rename!)
- ▶ press TAB to autocomplete names
- ▶ press up-arrow to go to previous command
- ▶ terminal is case-sensitive

Assignment 1: Overview

LING472

Section

Plan

My notes on the course

Logistics

CompLing and NLP

Technical overview

Assignment 0

Assignment 1

- ▶ Sample* concepts:
  - ▶ Programming: Execution flow, loop, variable, standard input (and regex of course)
  - ▶ Linguistics: Paradigms, agreement, case

*Not necessarily an exhaustive list

Regular expressions

- ► A module in python
- ► Can use debugger
- ► Do read the documentation of sub(), search() functions of re module and generally of any function you are trying to use
  - ► The language there may be annoying but those docs are complete and contain all the info you need
- ► Also, highly recommended:
  - ► https://regex101.com/