# Lecture 1: Course Introduction, Setup

Ling 1330/2330 Intro to Computational Linguistics Na-Rae Han, 8/27/2024

### Objectives

- Computational linguistics: what is, what we will learn
- Course Introduction
  - Syllabus: <u>https://sites.pitt.edu/~naraehan/ling1330/</u>
- Introduction to Python 3
  - Using **Python Interactive Shell** (ex. IDLE, IPython)
  - A warm-up: processing a string
  - Setup, housekeeping



### Language engineering & AI

- **Computational linguistics**: an interdisciplinary field dealing with modeling of <u>natural language</u> from a computational perspective.
- A "natural language"?
  - Any real-world human language.
  - cf. constructed, artificial, or computer languages such as: formal (logic) language, computer-programming languages, Klingon, etc.
- Areas: natural language processing (NLP), natural language understanding (NLU), human language technology (HLT), speech processing and synthesis, dialogue systems...

← part of AI (<u>Artificial Intelligence</u>)

# Role of linguists

- What is the role of linguists in all of these?
- Linguistic expertise intersecting with engineering is in high demand. Tasks include:
  - Build linguistic infrastructures: grammars, lexicons, ontologies (knowledge bank)
  - Manage language data: design and run linguistic annotation projects
  - Design dialogue systems for voice assistants
  - Consult for quality enhancement (machine translation systems, search results, etc.)
  - Build NLP systems, train models
  - Set standards and guidelines for localization

#### ← Working knowledge of computational linguistics is essential

# **Computational linguistics**

- What is the goal of computational linguistics?
  - Represent human language in terms of information processing.
- Theoretical computational linguistics
  - Investigates the computability of human language and linguistic theories.
  - Formal language theory and automata theory.

#### Applied computational linguistics

- The focus is on building (working) models of human language.
- Natural language processing (NLP) and engineering (NLE), human language technology (HLT), artificial intelligence (AI)
- Often centers around building practical applications: spell checkers, machine translation systems, automatic grader, etc.
- Two distinct flavors: <u>symbolic</u> vs. <u>statistical</u> approaches
  - Or maybe three now? + <u>neural</u> approach

### Computational linguistics course

#### Old LING 1330 course description:

In both linguistics and computer science, we need to study languages and their **grammar** from a **mathematical** point of view. This course is an introduction to **the mathematical theory of language and its applications**. The first half will deal mainly with elements of the theory of **automata** and its relation to grammars. The second half will survey ways in which this theory can be applied to English grammar and to the **design of programming languages**. We will concentrate on syntax, but will also pay some attention to theories of meaning.

• The class was about *theoretical* and *symbolic* computational linguistics.

# What we learn in *this* class

- Theoretical computational linguistics
  - Formal language theory
    - Automata, the Chomsky hierarchy
      - $\leftarrow$  We will cover only the basic ideas.
- Applied computational linguistics
  - Text and corpus processing
  - Using NLTK as an NLP library
  - Natural language processing applications:
    - Spelling correction, POS tagger, morphological analyzer/synthesizer, machine translation, text classification



### Syllabus time!

- Go over the course home page
  - https://sites.pitt.edu/~naraehan/ling1330/
  - Schedule, requirements, policies

### Python: script vs. shell

- **1. Python script** is a plain-text file with .py extension.
- Python shell is an interactive environment where Python will respond to each individual Python command.

#### ← Great tool for learning!

<pre>hello.py - C:\Users\Jane Eyre\Documents\</pre>	<pre>Python 3.8.3 Shell File Edit Shell Debug Options Window Help Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22 tel)] on win32 Type "help", "copyright", "credits" or "license()" &gt;&gt;&gt;</pre>
Python SCRIPT file	========= RESTART: C:\Users\Jane Eyre\Documents Hello, world! My name is Homer
Ln: 3 Co	(Python interpreter)

# Following up a script in shell

훩 hello.py - C:\Users\Jane Eyre\Documents\	—		×	Python 3.8.3 Shell
File Edit Format Run Options Window Help				
<pre>print('Hello, world!')</pre>			~	File Edit Shell Debug Options Window Help
name = 'Homer'				Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22
<pre>print('My name is', name)</pre>				tel)] on win32
				Type "help", "copyright", "credits" or "license()"
				======================================
				Hello, World!
				My name is nomen
				'Homer'
				<pre>&gt;&gt;&gt; print('Welcome.', name)</pre>
				Welcome, Homer
				>>>
		L pr	3 Col· 5	Variable name is
		L(1.	5 COI. 5	
				available in shell for
				subsequent commands
				subsequent commands

Warm-up: fox in sox

Let's practice string processing.

- Download this template script:
  - https://sites.pitt.edu/~naraehan/ling1330/fox in sox.TEMPLATE.py
  - ← Will paste link on MS Teams
- ▶ Complete [1] [5].



### Speed-coding fox\_in\_sox.py

- Video of me completing the script, in 5x speed!
  - https://sites.pitt.edu/~naraehan/ling1330/code\_in\_shell.mp4
- Watch closely! This is an IMPORTANT video: It will change your life.

### How to program efficiently

### The rookie way:

- You are doing all your coding in the script window, blindly.
- You only use the IDLE shell to verify the output of your script.

🔁 C./Users/zoso/Desktop/pythoncode/fox.py (3.5.1)	Python 3.5.1 Shell
File Edit Format Run Options Window Help	File Edit Shell Debug Options Window Help
<pre>fox = """Through three cheese trees three free fleas flew.</pre>	There are 37 words in the text.
While these fleas flew, freezy breeze blew.	There are 4 lines in the text.
Freezy breeze made these three trees freeze.	What to search? fr
Freezy trees made these trees' cheese freeze.	5
That's what made these three free fleas sneeze."""	>>>
print(fox)	====== RESTART: C:/Users/zoso/Desktop/pythoncode/fox.py ====== Through three cheese trees three free fleas flew. While these fleas flew, freezy breeze blew.
# [1] Print how many characters are in fox. YOUR CODE BELOW.	Freezy breeze made these three trees freeze.
print("There are", len(fox), "characters in the text.")	Freezy trees made these trees' cheese freeze.
	That's what made these three free fleas sneeze.
# [2] Print how many words are in fox. YOUR CODE BELOW.	There are 232 characters in the text.
print("There are", len(fox.split()), "words in the text.")	There are 37 words in the text.
# Fol Deist has seen lines and in few Your CODE DELOU	There are 4 lines in the text.
# [3] Print now many lines are in fox. YOUR CODE BELOW.	What to search? fr
print( There are, fox.count( (n ), Times in the text. )	7
+ T	>>>
"L	====== RESIARI: C:/Users/zoso/Desktop/pythoncode/fox.py ======
Vou are doing all	Inrough three cheese trees three free fleas flew.
<b># I UU AI C UUII Y AII</b> din	while these fleas flew, freezy breeze blew.
# 41	Freezy breeze made these three trees freeze.
the coding in the soft.	That's what made these three free fleas speeze.
	There are 332 characters in the text
corint window	There are 37 words in the text.
Script willdow	There are 4 lines in the text.
	What to search? fr
	Three are 7 tokens of fr
	>>>
Ln: 24 🤆 : 0	Ln: 114 Col: 4
	99% 📽 🔺 🍽 🔐 3/48 PM

# How to program efficiently

#### The seasoned Python programmer way:

- After a script is run, all the variables and the objects in it are still available in IDLE shell for you to tinker with.
- Experimenting in SHELL is much quicker – instant feedback!
- You should go back and forth between SCRIPT and SHELL windows, successively building up your script.

🛃 fox2.py - C:/Users/zoso/Desktop/pythoncode/fox2.py (3.5.1)	- 0 %	Python 3.5.1 Shell	
File Edit Format Run Options Window Help		File Edit Shell Debug Options Win	dow Help
<pre>fox = """Through three cheese trees three free fleas</pre>	flew. 🔺	>>> print(fox.count(what	at))
While these fleas flew, freezy breeze blew.		5	
Freezy breeze made these three trees freeze.		$\rightarrow$ fox.lower()	
Freezy trees made these trees' cheese freeze.		"through three cheese t	trees three free fleas flew \nwhile these
That's what made these three free fleas sneeze."""		fleas flew freezy bree	aze hlew \nfreezv breeze made these three
		trees freeze \nfreezy	trees made these trees' cheese freeze \nth
print(fox)		at's what made these th	nee free flees sneeze "
		the swind made chese ch	what)
# [1] Print how many characters are in fox. YOUR COD	E BELOW.		wina cy
print("There are", len(fox), "characters in the text	.")	() nnint(fox lowon()	count(what))
	. ,	>>> print(+ox.iower().0	count(what))
# [2] Print how many words are in fox. YOUR CODE BEL	DW.	/	(av lever() exact (whet) "takens ")
print("There are" len(fox split()), "words ")		There are ,	Tox. Lower(). count(what), tokens.)
princ( mere are ; ich(toxtopiic()); wordor )		There are 7 tokens.	
# [3] Print how many lines are in for VOUR CODE BEL	лы	>>> print("Inere are",	<pre>tox.lower().count(wnat), "tokens of", wna</pre>
<pre>mrint("There are" for count("\n") "lines ")</pre>		t)	-
princ( mere are, fox.counc( (n ), fines. )		There are / tokens of -	Fr
# [4] Promot for user input VOUR CODE RELOW		>>>	
what = input("What to search? ")		====== RESTART: C:/Us	
what = input( what to search? )		Through three cheese	
# [5] Dwint hav many times the year symplical stains	to found in .	While these fleas fle	Most codina
# [5] Print now many times the user-supplied string .	is tound in .	Freezy breeze made th	
# Make sure case is ignored. YOUR CODE BELOW.	<b>CH</b> 1 13	Freezy trees made the	activity hannone
print("There are", fox.lower().count(what), "tokens of	of", what)	That's what made the	activity nappens
		There are 232 charact	
		There are 37 words.	in SHELL
		There are 4 lines.	
		What to search? fr	
		There are 7 tokens of	
(		>>>	1
	Ln: 24 Col:		cm 118 10
			3:55 PM
			99% 🍽 🔺 🏲 🔐 🗤 9/8/2010

# Command history

- Every command you type in is stored as command history.
- You can quickly <u>bring up and edit</u> previously entered commands using these shortcuts (IDLE):
  - Windows: Alt+p / Alt+n
  - Mac: Ctrl+p / Ctrl+n (previous/next)



But these key combos are cumbersome! Let's remap to:

- (Up arrow: **previous** command)
- (Down arrow: next command)

Arrows are commonly used in shell



Remap "history-next" to down arrow, 3. and "history-previous" to up arrow

X

# Housekeeping (1)

- All installation & setup instructions are on "Checklists" page: <u>https://sites.pitt.edu/~naraehan/ling1330/checklists.html#setup</u>
- Install Python, version 3.11 or 3.12.
  - Anaconda Python recommended (details on checklists page!)
  - Also good: official python.org distribution
- Launch IDLE shell.
  - Anaconda users: learn how to launch it
  - Windows Anaconda users: need some config





# Housekeeping (2)

#### Designated folder for coding work:

- Create a folder within Documents called "ling1330" or "ling2330", and store all your coding work there.
- So this folder will be something like:
  - \* C:\Users\yourname\Documents\ling1330 (Windows)
  - \* C:\Users\yourname\OneDrive\Documents\ling1330 (Windows 11)
  - /Users/yourname/Documents/ling1330 (Mac)
- Verify your default "current working directory":

Python 3.8.3 Shell				
File Edit Shell Debug Options Window Help	Good	d! sensible default CWD,		
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22	: 2	points to user folder		
Type "help", "copyright", "credits" or "license()"	for mor	or mor		
<pre>&gt;&gt;&gt; import os &gt;&gt;&gt; os getcwd()</pre>	NOT good: system directories like			
'C:\\Users\\narae\\Documents'	C:\Program Files\Python311			
>>>		18		

### Wrap-up

- Exercise #1 out: Python refresher quiz
  - Due 30 minutes before next class (9am Thursday), on Canvas
- Take Day 1 Survey
  - On Canvas.

#### Help with your Python settings?

Come see us. Na-Rae: Wed 1-3pm, Teresa: Thu 5-8:30pm (temporary office hours for this week)

#### Next class (Thu):

Encoding systems, writing structured programs