Natural Language Processing

Dr. Karen Mazidi

NLP *aka* Human Language Technologies

Dr. Karen Mazidi

Karen.Mazidi@utdallas.edu

972-883-3868

ECSS 3.203

Office hours:

- Monday, Wednesday 5:30 – 6:00 pm
  - or by appointment

Agenda

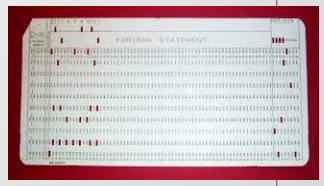
- Introductions
- Look at course content and resources
- Syllabus
- Chapter 1 of the book (brief overview)
- Quick look at Python

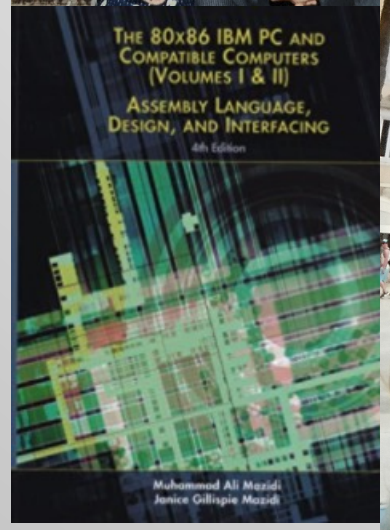
How I got here . . .

- My life in a slide

How I got here . . .


```
PROGRAM math
!Simple Fortran program to take in some variables and
!perform basic !mathematical operations.

!Explicitly define all variables:
Implicit None

!Variable declarations:
Real :: x, y, a, f1, f2, f3
Integer :: b,c

!Assign numerical values:
x = 1.1; y = 2.5; a = -5.5; b = 10; c = 3;

!Add, subtract, multiply and divide some numbers:
f1 = (x+y)/y
f2 = (a*b) + (c-a)
f3 = (x-a)/(-a*y)

!Print results to terminal:
Print*,"x=",x
Print*,"y=",y
Print*,"a=",a
Print*,"b=",b
Print*,"c=",c
Print*,"(x+y)/y=",f1
Print*,"(a*b) + (c-a)=",f2
Print*,"(x-a)/(-a*y)=",f3

STOP;
```





Hello

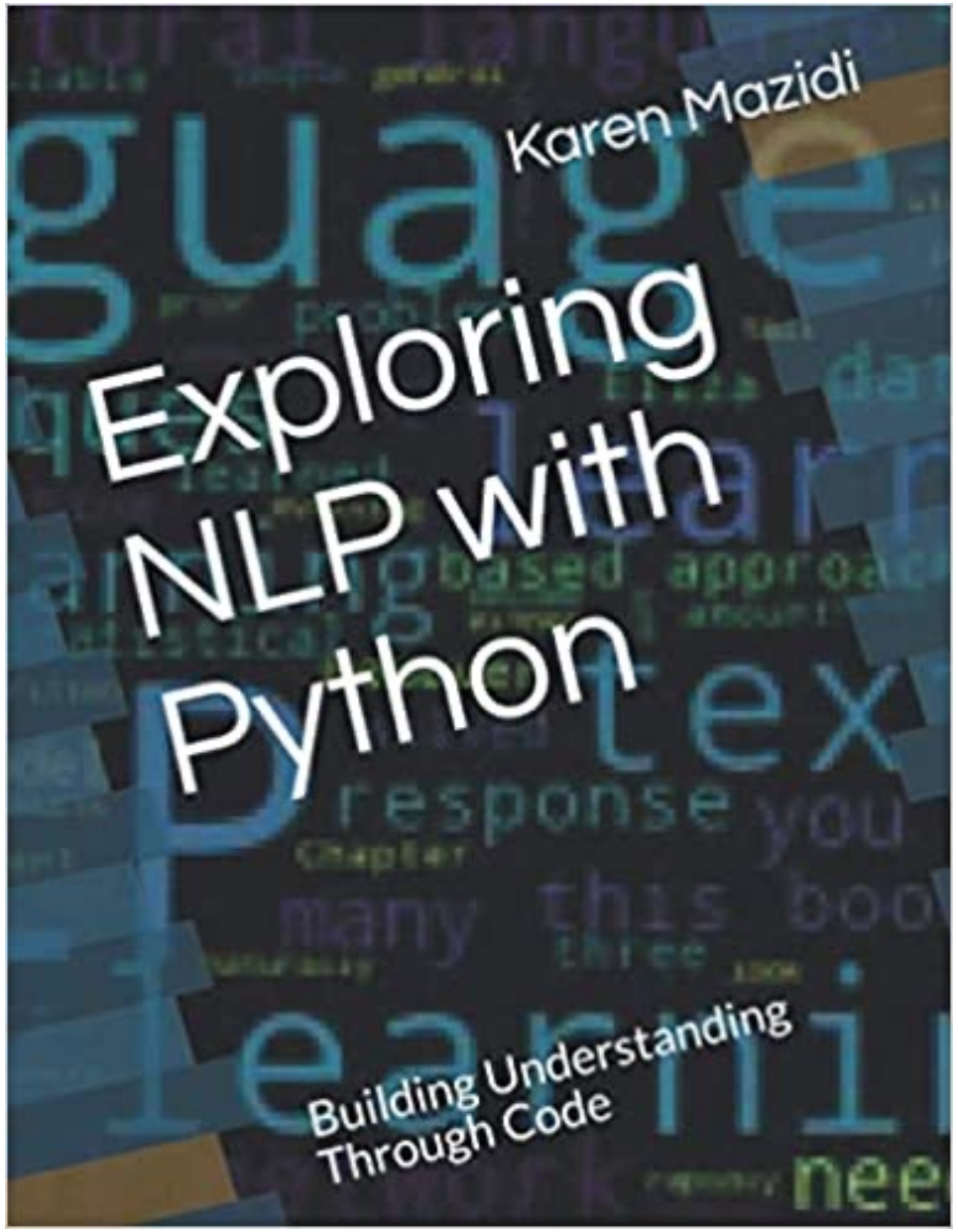
- Intro to course and each other

A look at the course content and resources

- eLearning – take the prerequisite survey
- GitHub https://github.com/kjmazidi/NLP
- YouTube https://www.youtube.com/c/JaniceMazidi

Our text

pdf available in eLearning



Project-Based Learning

- no exams
- less lecture, more student involvement in presenting material
- students strengthen collaboration and communication skills
- students showcase skills in a portfolio

Project-Based Learning

- The consensus among universities is that students should spend 2-3 hours of outside study/work for every hour of class time
- that is 5 to 7.5 hours for us
- Estimates for this class:
  - 1 hour read chapter
  - .5 hour take quiz
  - 4-5 hours/wk assignments
  - ++ hours for longer assignments
  - .5 hour for team meetings for group assignments

Project-Based Learning

What to expect from grading:

- A+ if you do all the work at an exemplary level
- A if you do all the work an an above-average level
- B if you do almost all the work at an above-average level
- C if you did 70% of the work at an average level
- D if you did 60% of the work at an average level

Natural Language Processing

Chapter 1

Chapter 1: Natural Language Processing

- Allows computers to “process” natural human language
  - OK Google
  - Identifying spam emails
  - Sentiment analysis
  - More applications every day

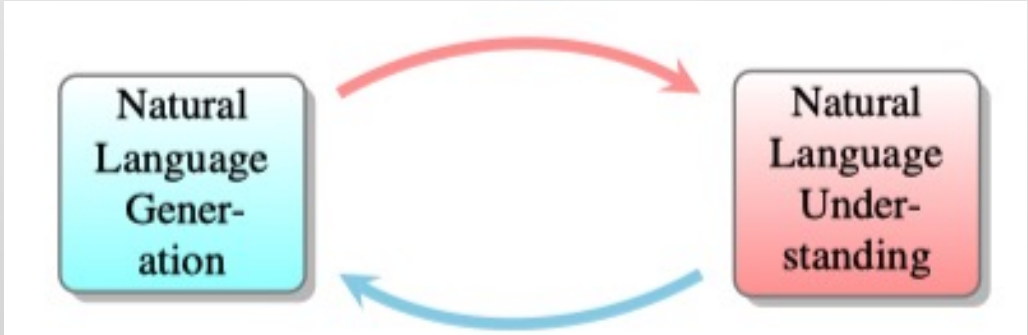
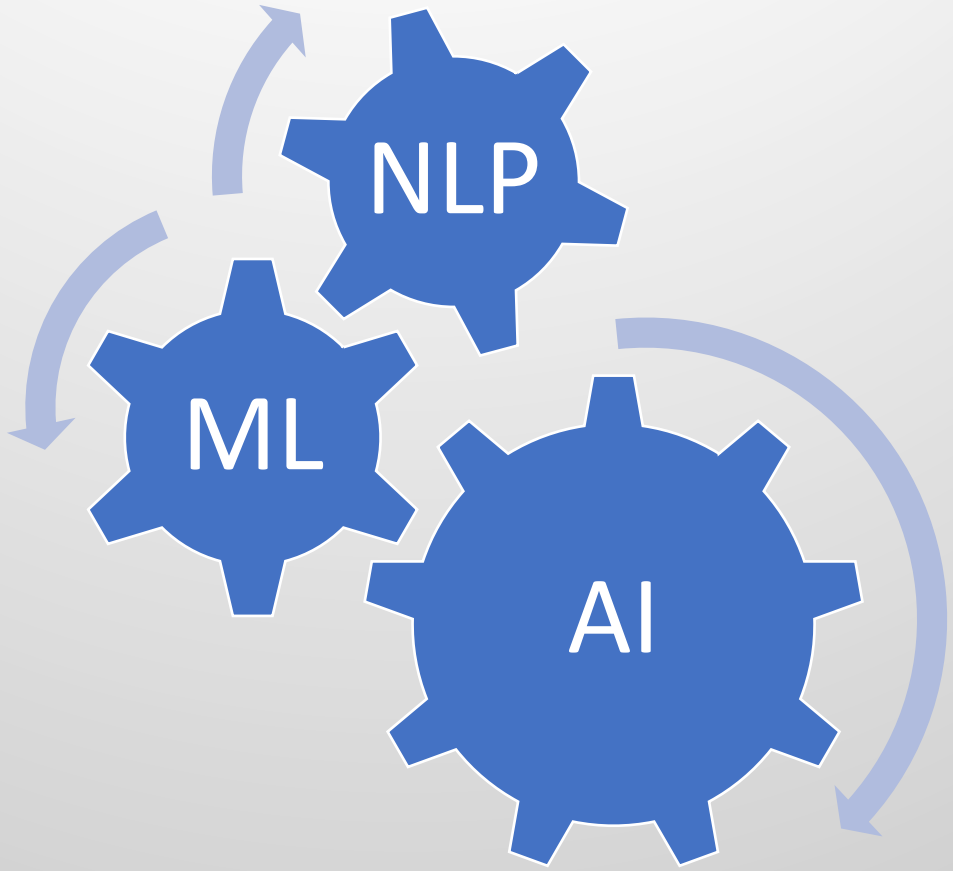
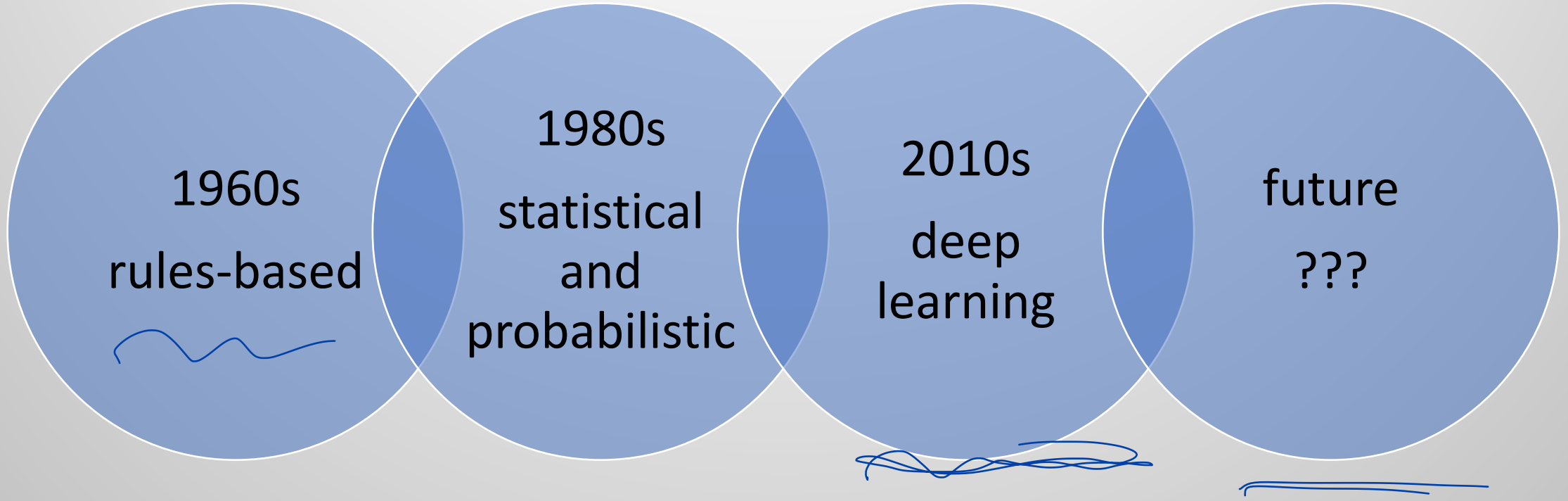


Figure 1.2: Natural Language Processing

NLP, ML, and AI



NLP approaches through time



Rules-based approaches

- Examples:
  - spell check
  - context-free grammar
  - Eliza chatbot (regex and a little code)
- Problem: rules don’t scale up to the complexity of human language

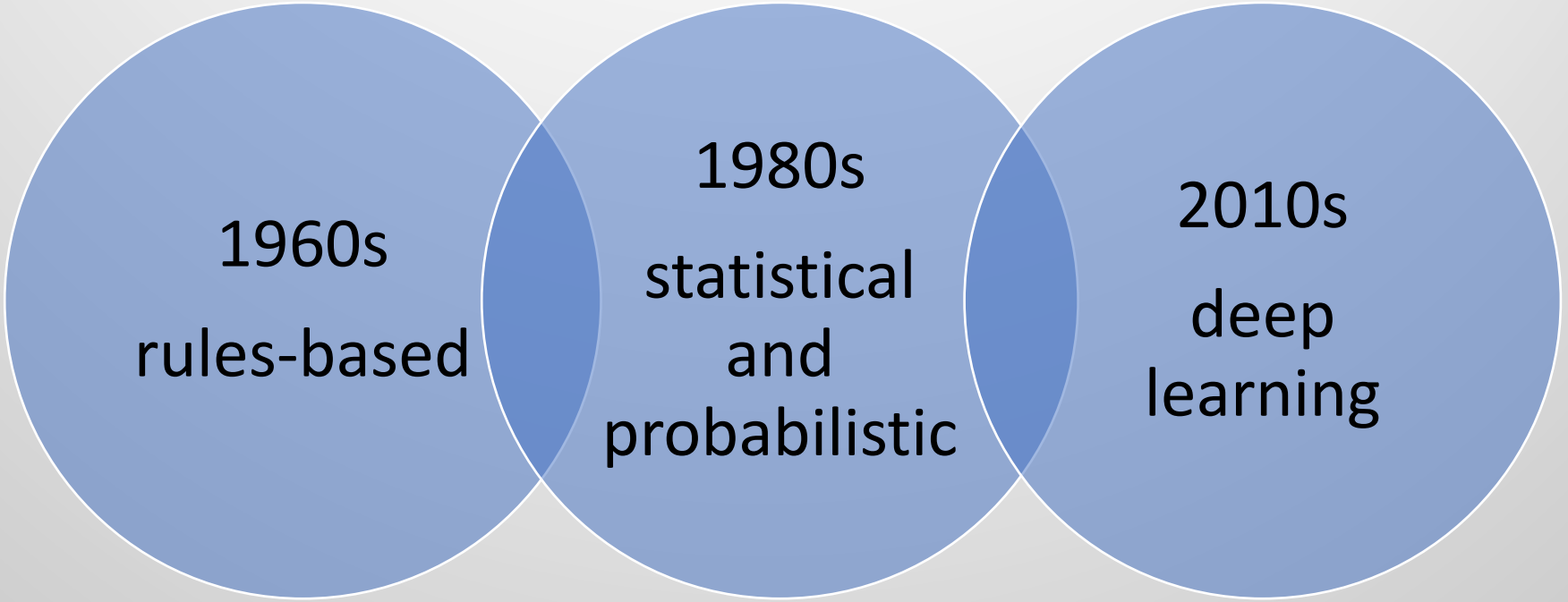
Statistical and Probabilistic approaches

- Examples learn from a corpus of data:
  - use of word frequencies
  - traditional machine learning algorithms
- Problem: need a moderate amount of data and good processing power

Deep Learning

- Improved results in:
  - language translation
  - language generation
  - language understanding
  - much more
- Problem: need a large amount of data and powerful processing
- Problem: hype

Most projects are a combination of these approaches



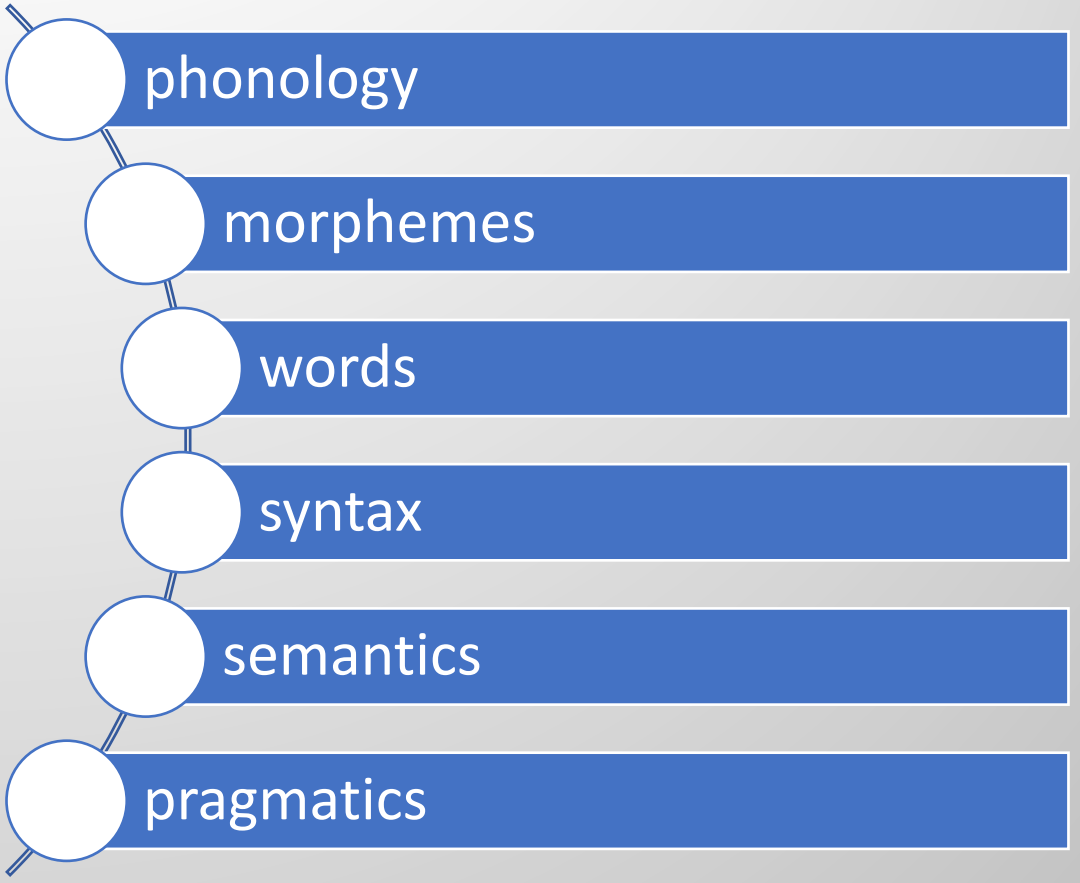
NLP or HLT?

- Natural Language Processing is the general term
- Aka Computational Linguistics
- Aka Human Language Technologies (at UTD)

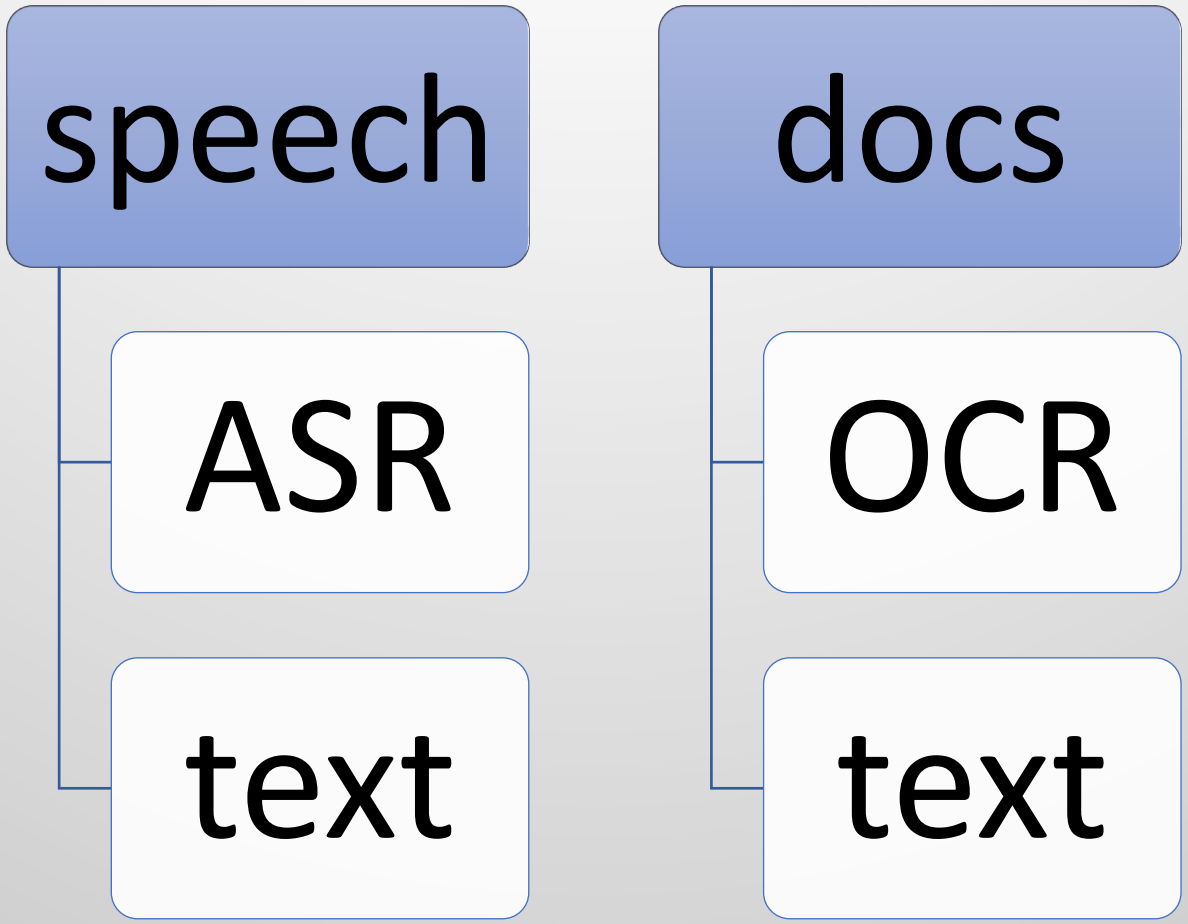
- The “natural” in NLP distinguishes between human language and formal languages like context-free grammars or programming languages

Language

- Linguists study many levels of language



Input to NLP systems

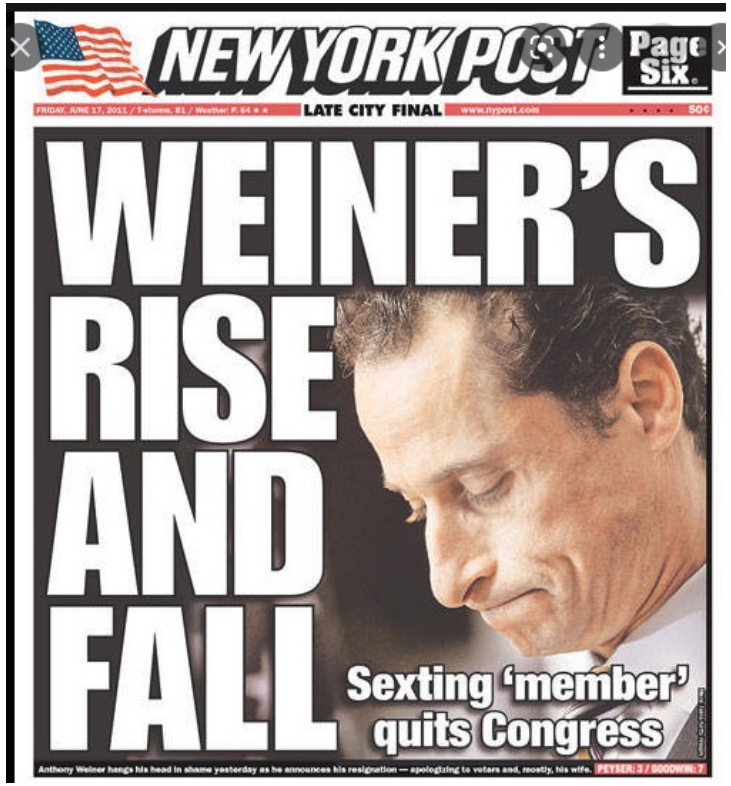


Why is NLP hard?

- Human language is vague, ambiguous
- Idioms and figurative speech
- Human conversation assumes real-world knowledge
- Human conversation assumes known context


JAY CRONLEY
One-armed man applauds the kindness of strangers
The other afternoon, a man and a


Marijuana issue sent to a joint committee


NEW YORK POST
Page Six
LATE CITY FINAL
WEINER'S RISE AND FALL
Sexting 'member' quits Congress
Anthony Weiner hangs his head in shame yesterday as he announces his resignation — apologizing to voters and, mostly, his wife.

headlines

Python

Python

- The most widely used language in NLP
- Next time: Python tutorial, feel free to bring laptop (not required)

- if confident-in-python:
  - Watch python review video
- else:
  - Watch python fundamentals videos

Install Python on your machine

- https://www.python.org/downloads/
- Version 3.7 or higher
- If you have Python 2, don’t delete it

Ways to run Python code

- At the Python terminal
- Run a Python script from system terminal
- Use an IDE (recommended for next few homework assignments) like PyCharm or the new DataSpell
- Use Jupyter lab on your machine
- Use Google Colab in the cloud (best for machine learning, later)

Changes from Python 2 to Python 3

- Many important changes under the hood
- Most noticeable change:

```
print "hello"   # Python 2
print ("hello") # Python 3
```

Things to remember about Python

- Python is an interpreted language; source code is compiled into byte code to be executed by the operating system
- All data items are objects
- Python uses dynamic duck typing
- Python uses indents to create code blocks, not { }
- The end of the line is the end of a statement, no ; needed
- Python is case sensitive
- Comments start with #

Style

- Spaces v. tabs
- 4 spaces preferred; Many IDEs replace tabs with 4 spaces
- camelCase v. underscores
- Underscores for variable and function names
- The authority is the PEP8 style guide: https://www.python.org/dev/peps/pep-0008/

- For this class, just be consistent

Python data

- The native data types in Python include:
  - int - non-limited length
  - float - same as C double
  - complex
  - boolean
  - string - ‘single’ or "double" quotes

- Python also has some useful built-in constants:
  - True
  - False
  - None

Python variables

- Think of Python variables like pointers to memory locations
- The “type” of a variable is the type of data that is currently in it
- Python doesn’t complain when you change its type

```
>>> v = 5
>>> type(v)
<class 'int'>
>>> v = 'a'
>>> type(v)
<class 'str'>
```

Python variables

- Python throws an error if you try to do something that is not compatible with the current type

```
v = 5
v += 1
print(v)   # v is now 6
v = 'a'
v += 'b'
print(v)   # v is now 'ab'
v += 1    #  will cause an error, should be v += str(1)
```

Console I/O Examples

```
name = input("What's your name? ")
print('Hello ', name, '!')  # notice that ',' adds a space
print('Hello '+ name + '!') # '+' (concatenate) does not add a space
```

The input() function reads whatever the user types as a string. To get numeric data from a user, the string input can be converted to numbers with the built-in int() and float() functions:

```
radius = input("Enter radius: ")
radius = float(radius)
area = radius * radius * 3.14
print("area = ", area)
```

Python scripts

- Plain text files that end in .py
- Python executes every statement at the leftmost outdent
- Example using command line system args:

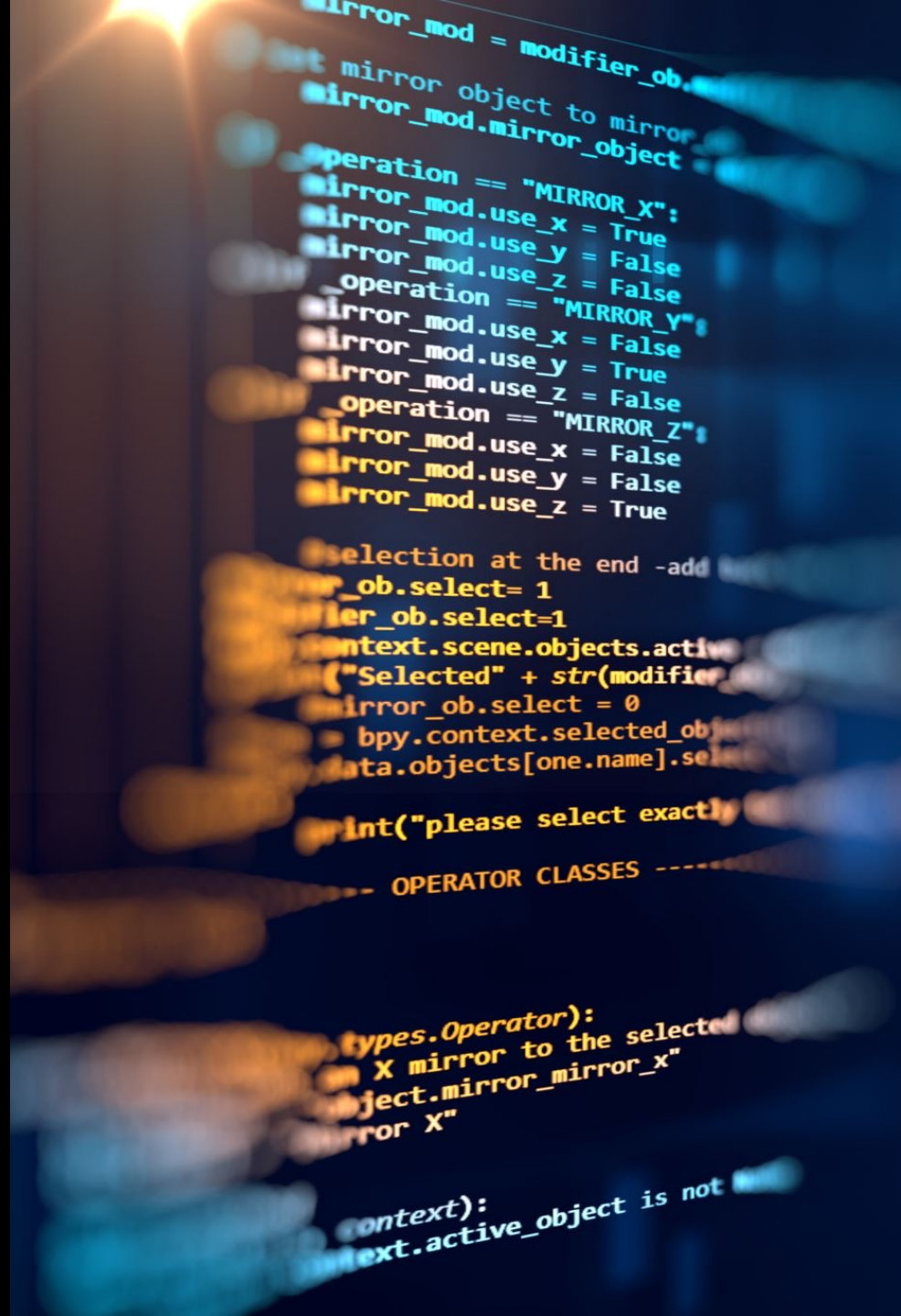
```
$python hello.py Karen
$Hello Karen!
```

Code 2.4.1 — A Python Script. Example with Arguments

```
import sys

def main():
    print("Hello " + sys.argv[1] + "!")

if __name__ == '__main__':  # uses double underscores (dunders)
    main()
```

Course Topics (and Book Parts)

- Part 1: Foundations
  - Python, NLTK, Linguistics 101
- Part 2: Words
- Part 3: Sentences
- Part 4: Documents
- Part 5: Machine Learning Approaches
- Part 6: Deep Learning for NLP



To Do

- Download the book pdf from Piazza
  - Read Chapter 1
- Take pre-requisite survey in eLearning
- Set up your Python programming environment
- Play around with Python (see Chapter 2)
- Next class is a Python tutorial

questions?

don’t be shy