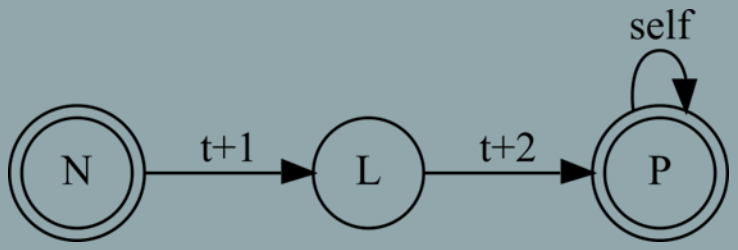
LING-362

Introduction to Natural Language Processing

Introduction

amir.zeldes@georgetown.edu



Organization

- Instructor: Amir Zeldes
  - amir.zeldes@georgetown.edu
  - Poulton 243
  - Office hours: Wednesdays, 15:30-17:00
- TAs:
  - Yang (Janet) Liu, yl879@georgetown.edu
  - Yilun Zhu, yz565@georgetown.edu
  - Office hours: TBA

- For questions about homework please remember to contact us on time!

Resources

- Many great books; we will use:
  - Bird, S., Klein, E., & Loper, E. (2017). *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly. (NLTK 3.2.4)
  - Jurafsky, D., & Martin, J. H. (2017). *Speech and Language Processing.* Upper Saddle River, NJ: Prentice Hall.
  - Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan Claypool.
- A very comprehensive Python reference if needed:
  - Lutz, M. (2013) *Learning Python*. Sebastopol, CA: O'Reilly
- Readings, slides, assignments and code on **Canvas**

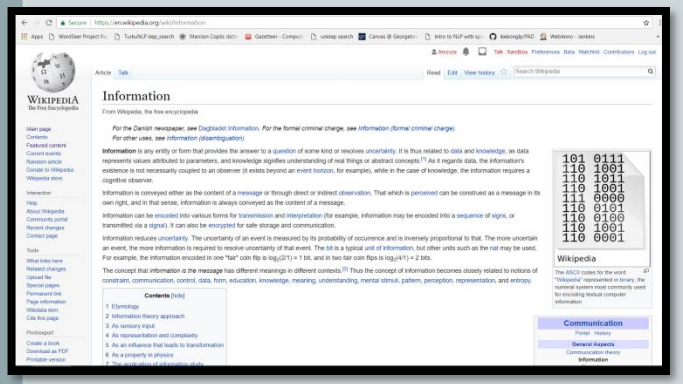
Resources

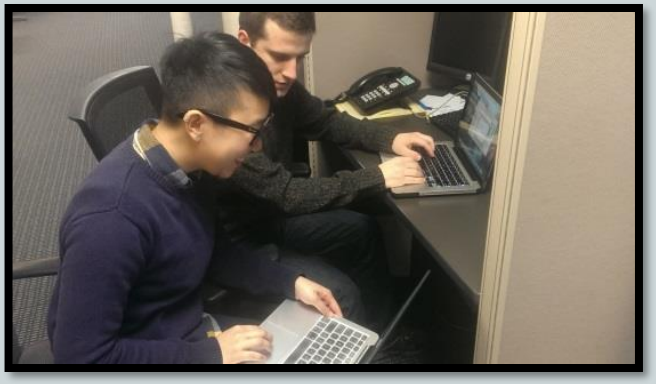
- Events and groups in the area:
  - CL community at Georgetown: http://gucl.georgetown.edu
  - DC NLP: https://www.meetup.com/DC-NLP/
  - Data Community DC: http://www.datacommunitydc.org/
  - DCPython: https://dcpython.org/ (Sep 21 – Evening of Python)
  - CLIP Talks at College Park: (can subscribe to list) https://wiki.umiacs.umd.edu/clip/index.php/Events
  - CLSP Seminars at JHU https://www.clsp.jhu.edu/seminars/action~month/exact_date~1630468800/cat_ids~730,731,732/request_format~json/
- Other mailing lists:
  - Send Nathan Schneider an e-mail for the GUCL list
  - Send me an e-mail for CorpInfo: corpinfo@georgetown.edu (announcements about new corpora)

Introductions

What is this course about?

- We are surrounded by computers in daily life
  - We communicate using computers
  - Find information on the Internet
  - Navigate using our cellphones
  - Spend a significant part of our lives working on a computer





What is this course about?

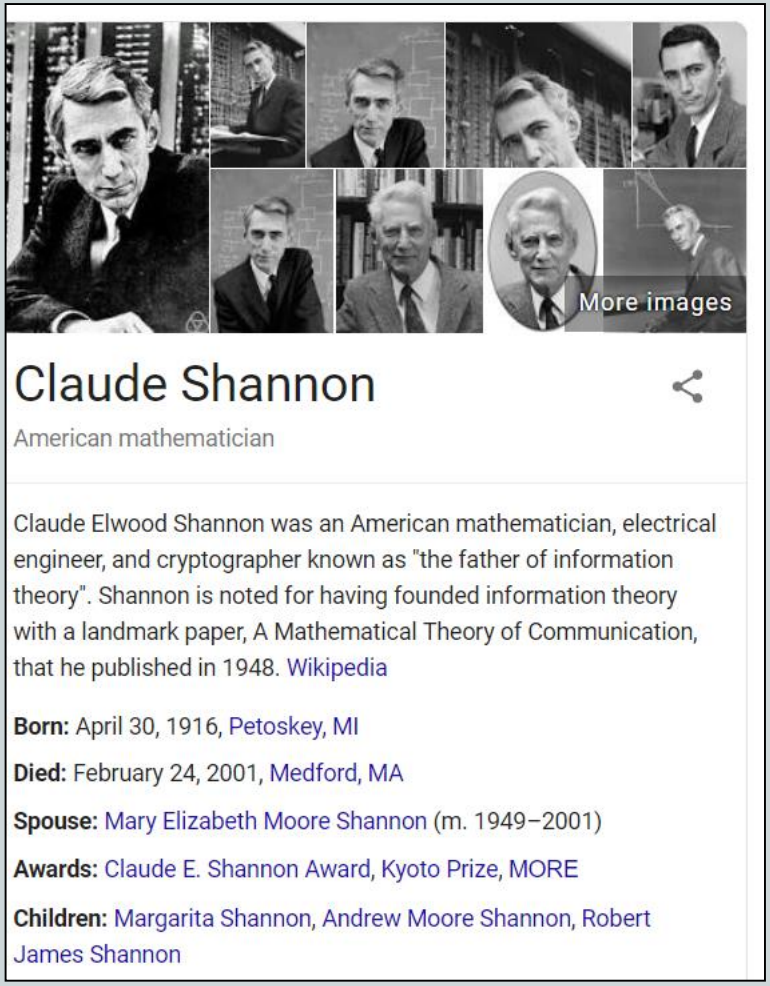
- But we don't speak the same language as computers
  - Computers run programs
  - Operate on **structured data**
  - Deterministic, do what they're told using some **foreseen** functions
  - No button like this:

Structured and unstructured

- **Structured data** is great for computers
  - Think about an abstraction representing vital statistics of a human being:
    - `name = (string: last, first) "Shannon, Claude"`
    - `date_of_birth = (date: yyyy-mm-dd) 1916-04-30`
    - `nationality = (string) "American" (possible values:195)`
    - `height = (numeric, in cm) 177`
    - `...`
- We can enter these into a database

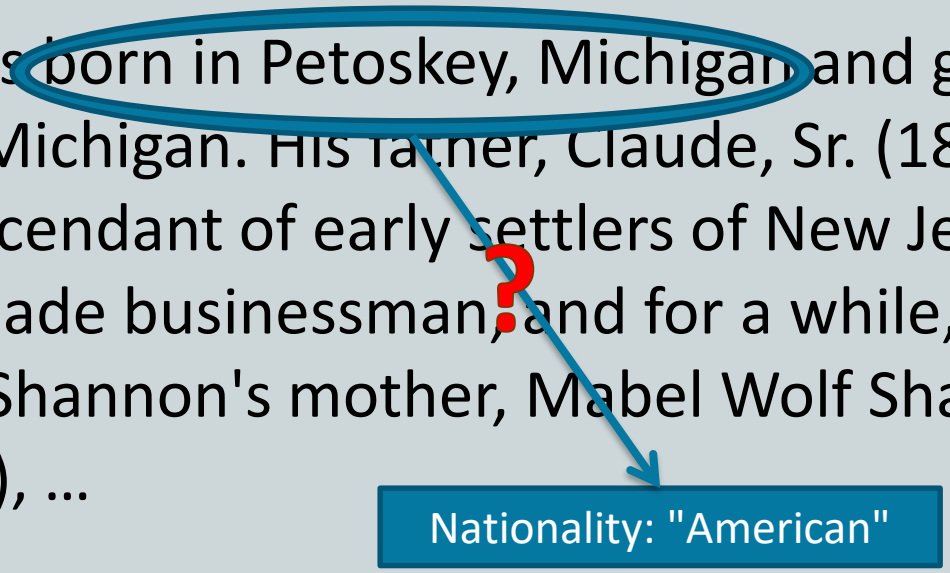
Structured data

- If we then Google Claude Shannon…
  - Matches a known "name" (=Named Entity Recognition, **NER**)
  - Structured information available
  - Entered into fields for display



Unstructured data

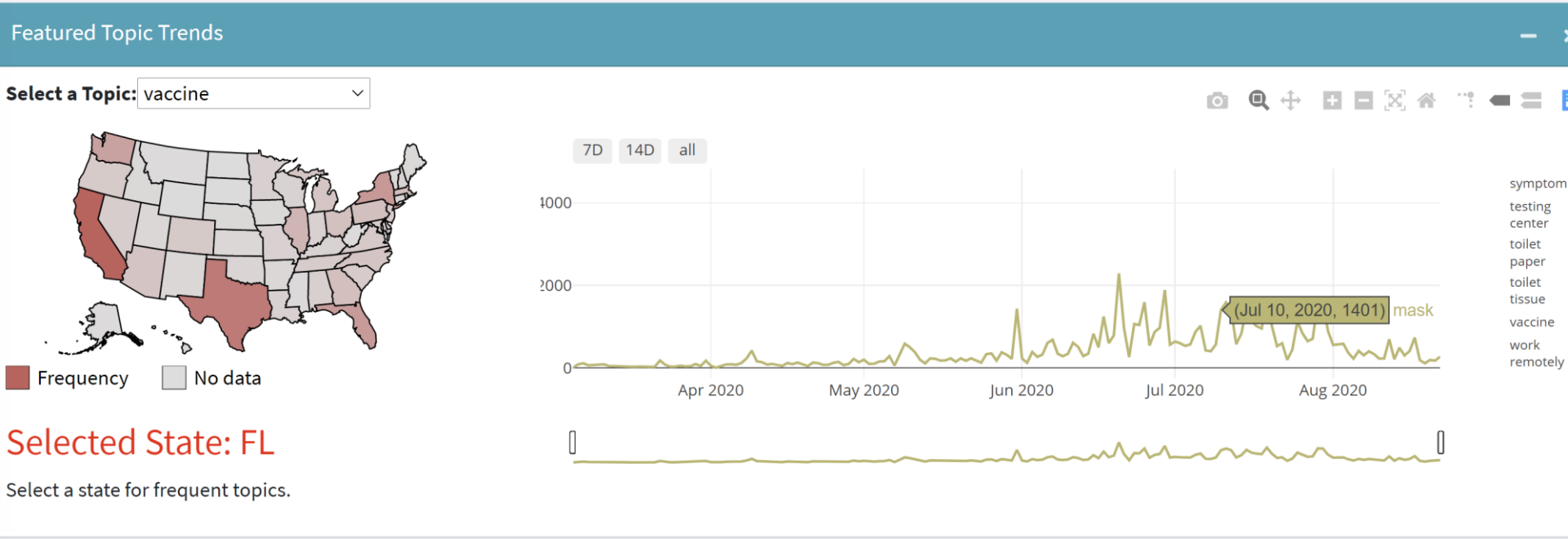
- We have **vastly** more unstructured* data
- It looks more like this:
  - Shannon was born in Petoskey, Michigan and grew up in Gaylord, Michigan. His father, Claude, Sr. (1862–1934), a descendant of early settlers of New Jersey, was a self-made businessman, and for a while, a Judge of Probate. Shannon's mother, Mabel Wolf Shannon (1890–1945), …



* Unstructured **language** data – there's even tons more of telemetry data out there…

Multiply by 100000000000....

- Frequency changes in COVID19 related terms on US Twitter by state



Source: https://mykabir.github.io/coronavis

The goal

- Computers should understand everything we say, write, etc.
  - We should be able to ask them questions in plain English (or French, or Wolof, or Cherokee, …)
  - They should be able to answer in plain English etc.
- If they could do that…
  - Remember that computers can stay up all night reading the entire Internet – they would know a lot!
  - But will they always be ‘right’? How would we know?
  - Why is this so hard? What have people been doing?

Exercise – anaphora resolution

- Download ***IRS_coref_ex.pdf*** from Canvas
- Notice the forms highlighted in the text
- Imagine you're a computer
  - What are the cues to the correct solution?
  - How do you operationalize them?
  - What kinds of mistakes might you make?

Atlas / Boston Dynamics

What would a computer need to know?

- Example: **anaphora resolution** & **NER**
- How do we know what this word refers to:

xxxxxxxxx xxxxx xx xxxxxxxx **its**

xxxxxxxxxxx xxxxxx xxxx

What would a computer need to know?

- Example: **anaphora resolution** & **NER**
- How do we know what this word refers to:

xxxxxxxxx xxxxx to improve **its**
xxxxxxxxxxx xxxxxx xxxx

What would a computer need to know?

- Example: **anaphora resolution** & **NER**
- How do we know what this word refers to:

xxxxxxxxx Miami to improve **its**
xxxxxxxxxxx xxxxxx xxxx

What would a computer need to know?

- Example: **anaphora resolution** & **NER**
- How do we know what this word refers to:

```
persuade Miami to improve its
xxxxxxxxxxx xxxxxx xxxx
```

What would a computer need to know?

- Example: **anaphora resolution** & **NER**
- How do we know what this word refers to:

`persuade Miami to improve` **`its`**
`city-owned xxxxxx xxxx`

What would a computer need to know?

- Example: **anaphora resolution** & **NER**
- How do we know what this word refers to:

`persuade Miami to improve` **`its`** `city-owned Orange Bowl`

- **Miami** is most recent noun before **its**
- **Miami** is inanimate and singular
- **its** is an owner of N, and that N is **city-owned**
- The Orange Bowl is a **FACILITY** entity in a **LOCATION (CITY)**: **Miami**

NLP – historical overview

- In some ways, NLP was historically at the heart of computation

- How did people want to test whether a computer is intelligent?

  -> **Turing Test** (1950-1954)

- How to teach computers recursive patterns?

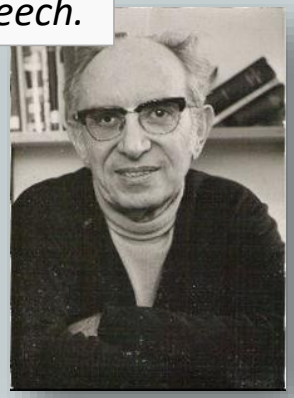
  -> Kleene's work on **regular languages** (1951)

NLP – historical overview

- **Machine Translation:**
  - Pioneered at MIT and Georgetown
  - Georgetown-IBM experiment (1951-1954), Russian-English

Мы передаем мысли посредством речи	*We transmit thoughts by means of speech.*

  - Subsequent euphoria quashed in 1966 advisory committee report



-- "fully automatic MT is not achievable in the foreseeable future"
(Bar-Hillel 1951 – one of the most misused quotes in CL!)

Bar Hillel's *pen* example

- What would you need to know to translate this:

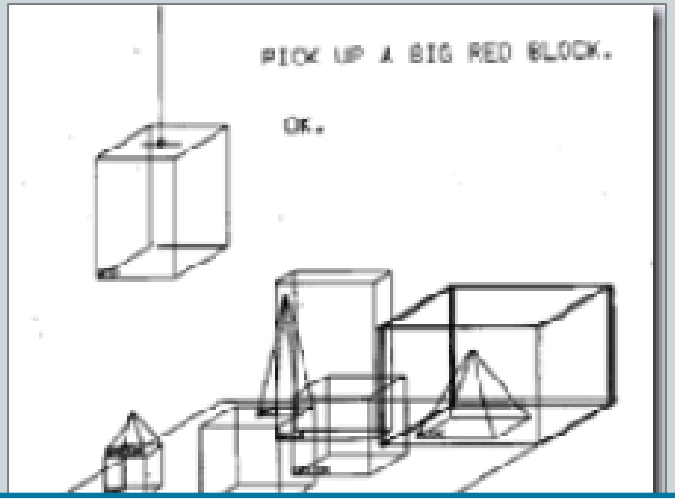
The box was in the pen

> "Whenever I offered this argument to one of my colleagues working on MT, their first reaction was: "But why not envisage a system which will put this knowledge at the disposal of the translation machine?" Understandable as this reaction is, it is very easy to show its futility. What such a suggestion amounts to, if taken seriously, is the requirement that a translation machine should not only be supplied with a dictionary but also with a universal encyclopedia. This is surely utterly chimerical and hardly deserves any further discussion." (Bar Hillel 1960)

Please read for next time!

60s-70s Symbolic approaches to NLP

- Computational symbolic work went hand in hand with the arrival of Generative Grammar
- Rationalist 'camp'
  - Context free grammars (Chomsky 1956)
  - Formal grammars for parsing (TDAP, Harris 1962)
  - Logic and rule based AI systems (SHRDLU, Winograd 1972)
  - Question answering…



Demo: http://hci.stanford.edu/winograd/shrdlu/download/javashrdlu.zip

60s-70s Stochastic approaches

- Simultaneously, growing interest in data driven learning methods
- Empiricist 'camp':
  - Character recognition with language model (Bledsoe & Browning 1959)
  - Authorship attribution – Federalist Papers (Mosteller & Wallace 1964)
  - Hidden Markov Models for speech recognition, tagging, noisy channel model for MT…

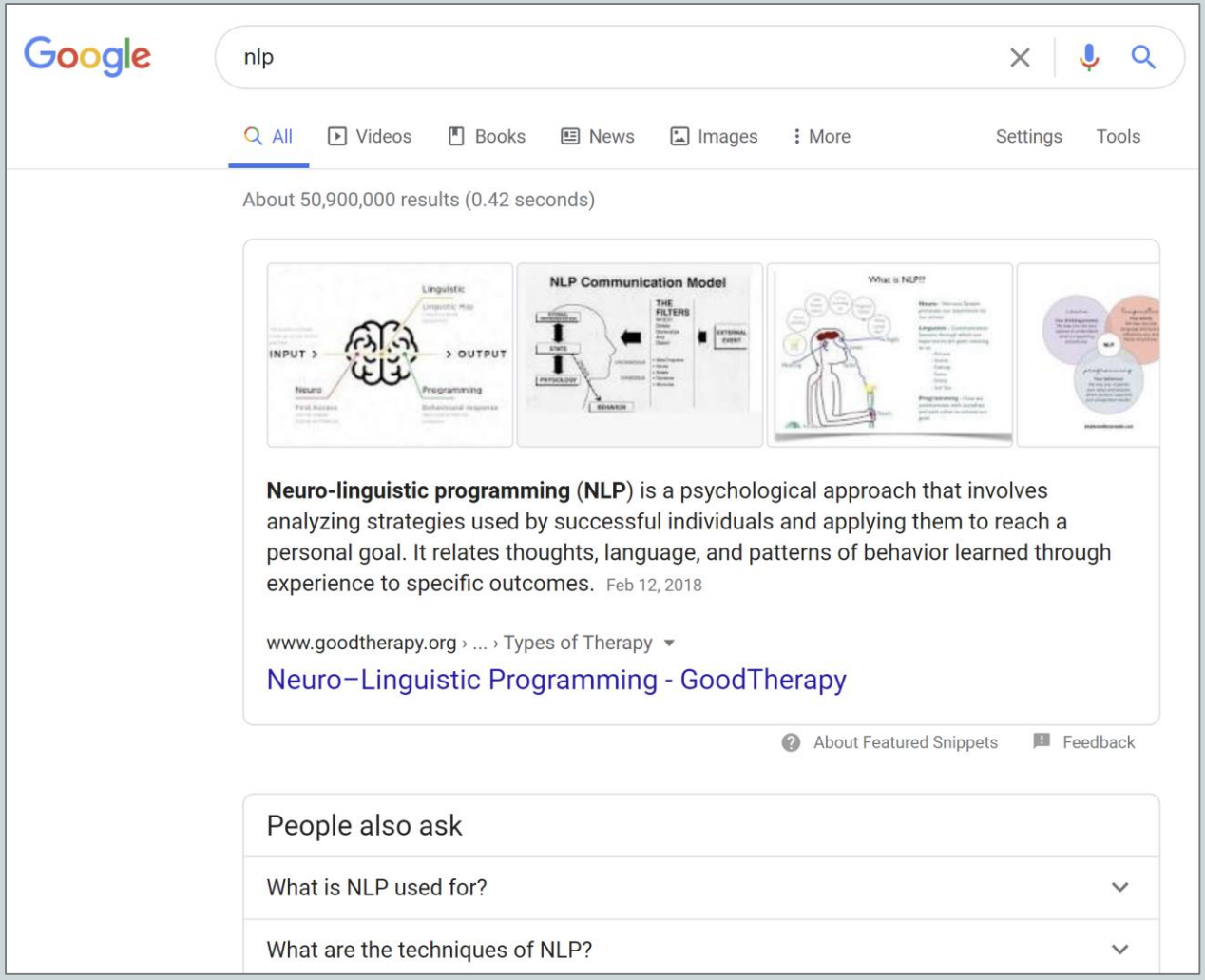
Coming together

- The following decades have seen stochastic and rule based approaches blend
- Much more data available:
  - Pioneering manual corpora of the 90s – Penn Treebank, PropBank, Penn Discourse Treebank
  - Lexical resources – Princeton WordNet, Berkeley FrameNet
  - And eventually, large amounts of Web data – Google n-grams, Google Books, WaCKy corpora, Gigaword corpus, Common Crawl…

What NLP can do today

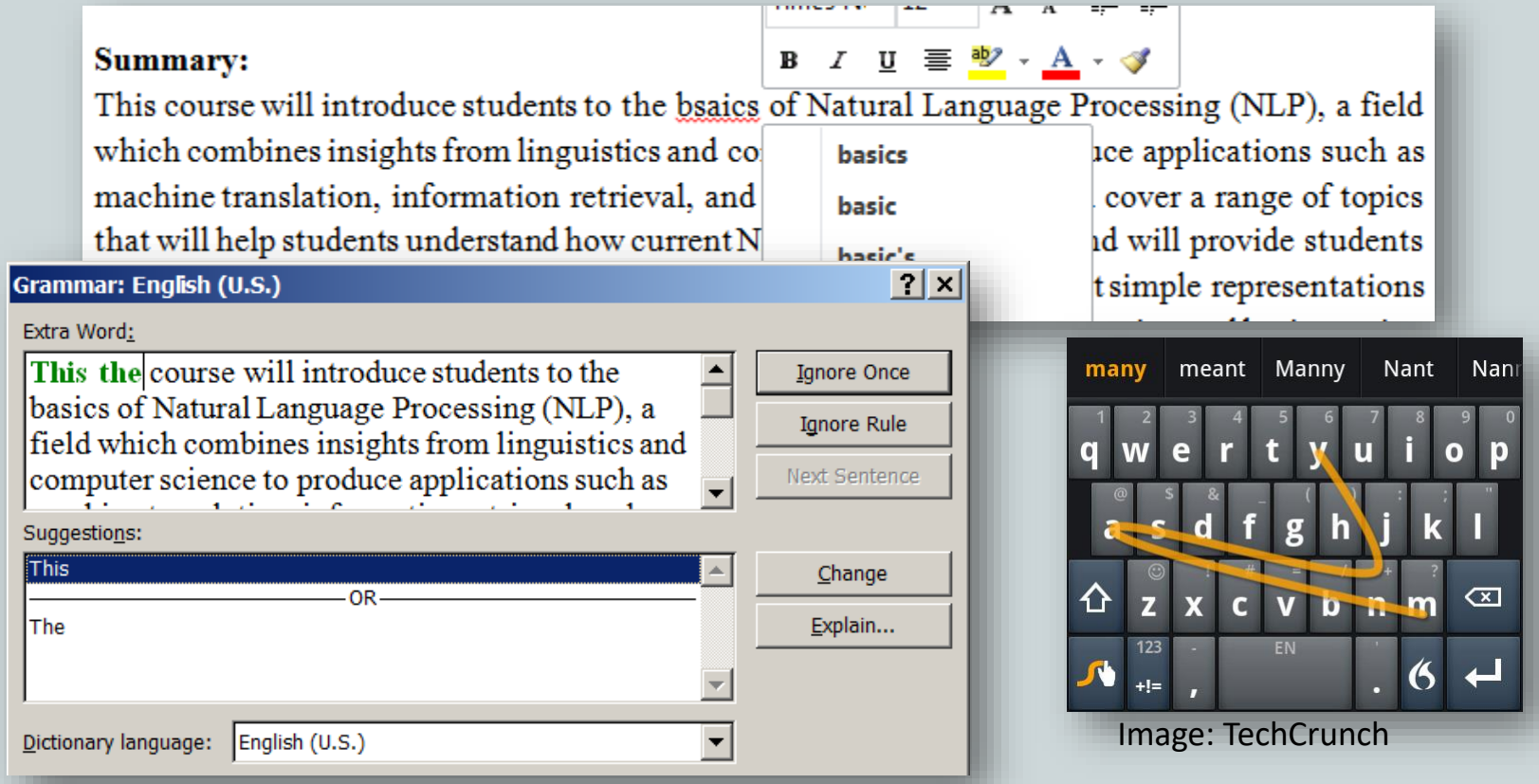
- NLP is about recognizing structure in unstructured text
  - components often chained in a **pipeline**
  - use cues from previous steps to recognize deeper categories (or do everything at once!)
- Give best guess given cues
  - be right as often as possible (=wrong as rarely as possible)
  - maximum likelihood, expectation maximization
- Select and recognize best features for task
  - use machine learning to choose and give weights to inputs

What NLP can do today



What NLP can do today

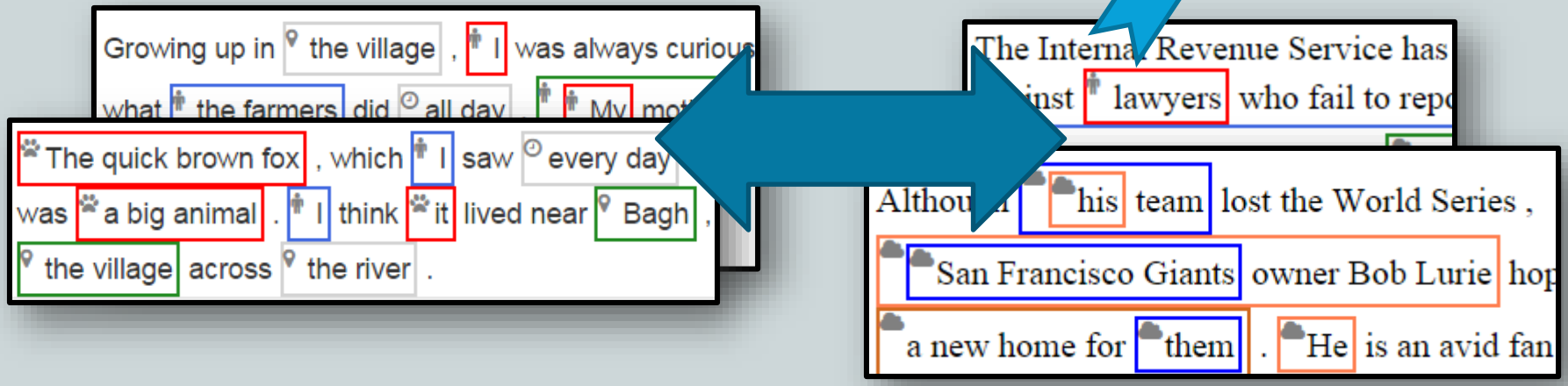
- Spellchecking, grammar, predictive keyboards



What NLP can do today

- Text analytics
  - Document classification
  - Knowledge extraction
  - Textual entailment

subject	predicate	object
Bagh	isa	village
Bob Lurie	owns	SF Giants
SF Giants	isa	team



What NLP can do today

- And more!
  - Speech recognition
  - Word sense disambiguation
  - Word and textual similarity metrics
  - Question answering
  - Natural language generation & understanding (NLG & NLU)
  - …

Programming in this course – Python

- We will be programming throughout this course with **Python**
- A highly accessible scripting language:
  - Interpreted language
  - Cross platform
  - No compilation – just run it and go
  - Supports object oriented programming (more later)
  - Highly extensible, fairly high performance

Downloading and installing

- Choose an installer for your platform:
  - https://www.python.org/downloads/release/python-395/
- Run Python from the command line terminal:

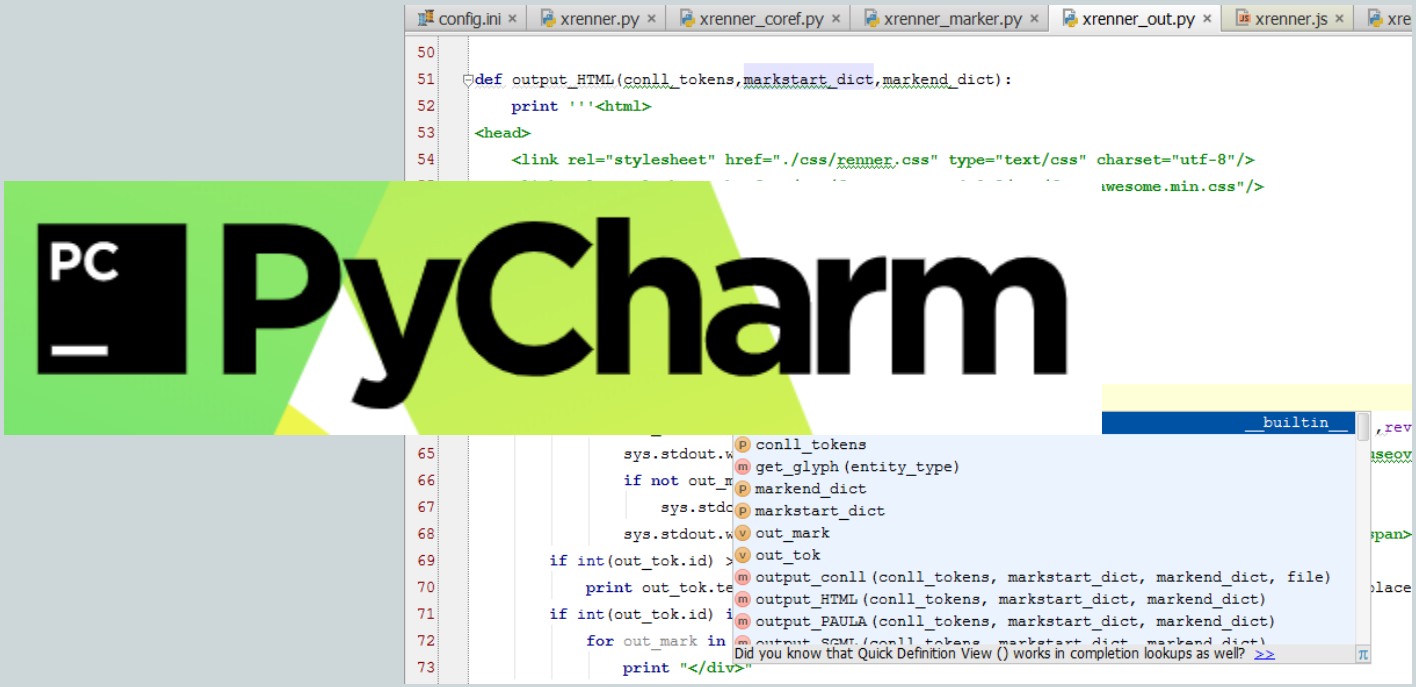
```
az364@LING-WL-2YW6Q13 c:\Users\az364\AppData\Local\Programs\Python\Python39
$ python --version
Python 3.9.5

az364@LING-WL-2YW6Q13 c:\Users\az364\AppData\Local\Programs\Python\Python39
$ python
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Integrated Development Environment

- The **IDE** is a development environment for your programming language
- Makes your life much easier
- Helps you learn and warns about problems

Integrated Development Environment

- We will use PyCharm, a cross-platform, freely available IDE (for non-commercial)
  - Download: https://www.jetbrains.com/pycharm/



Integrated Development Environment

- IDEs help you by:
  - Syntax highlighting (keywords like ***if***, ***else***, …)
  - Error checking
  - Debugging (run program step by step)
  - Keeping track of available variables and functions
  - Auto-completion
  - Formatting conventions
  - Compatibility inspections (Python 2.X, 3.X)
  - And much more!!

For next time

- Install Python 3.9 (64 bit if your machine supports it)
- Install PyCharm community edition (for non-commercial/academic)
- Submit Assignment 1 by Monday before class:
  - Installing Python
  - First math exercises
  - Questions about Bar Hillel (1960)

Want to start practicing?

- I recommend working through the Natural Language Toolkit **(NLTK)** book:
  - https://www.nltk.org/book/

- You'll need to install NLTK and run Python from your terminal…
- This is not part of the homework submission!

NLTK – Installing

- Download and install using instructions from http://www.nltk.org
- Or run in terminal:
  - ***sudo pip install nltk*** (Mac/Linux)
  - ***py -3.9 -m pip install nltk*** (Windows)
- Once installed, run Python in terminal
- Download resources:

```
import nltk
nltk.download()
```