Lecture 1

408/508 *Computational Techniques for Linguists*

Administrivia

1. Syllabus
2. Questions about the Syllabus
3. Introduction

- I will assume everyone has a laptop or desktop …

Syllabus

Description of Course

- An introductory level course to computers for linguists (and non-engineers).

Course Pre-requisites

- None!

Instructor and Contact Information

- Instructor: Sandiway Fong, Douglass 311.
- Contact email: sandiway@arizona.edu (all homework to be submitted here).
- Homepage: `sandiway.arizona.edu`
- Instructor: Sandiway Fong, Dept. of Linguistics Office: Douglass 311

Syllabus

Hours:

- make appointments by email or drop by my office
- ask after class (best for quick questions)

Meet:

- C E Chavez Bldg, Rm 405
- Tuesdays/Thursdays 12:30PM - 1:45PM

Course Format and Teaching Methods

- Lecture with slides.
- Panopto videos (when available) for lecture review.
- All homeworks will be introduced and reviewed in class.

Syllabus

Course Objectives

Topics covered include:

- Fundamental concepts
  - computer organization: underlying hardware, and operating systems (processes, shell, filesystem etc.)
- Operating System:
  - **Ubuntu** (Linux) and the **Terminal** (Shell usage and programming)
- Introduction to programming
  - data types, different programming styles, thinking algorithmically …
- Programming Languages:
  - *selected examples*: Bash shell, Python, Javascript, Perl, Tcl/Tk, HTML/CSS, cgi-bin etc.

Syllabus

Course Learning Outcomes

After completing this course, students will:

- be familiar with the underlying technology:
  - *What makes a computer tick*? *Why does it work that #@!&% way*?
- acquire the ability to think algorithmically
  - not necessarily the same as logic
- acquire the ability to write short programs
  - becoming a good programmer takes lots of practice (*and mistakes along the way*)
- build a graphical user interface (GUI)
- build a web application (with a relational database)
- be equipped to take classes in the Human Language Technology (HLT) program

Syllabus

Absence and Class Participation Policy

- I expect you to attend lectures (though attendance will not be taken).
- The UA's policy concerning Class Attendance, Participation, and Administrative Drops is available at: http://catalog.arizona.edu/policy/class-attendance-participation-and-administrative-drop.
- Tell me ahead of time so we can make alternative arrangements in the case of missed homeworks. **No homework will be accepted late. Explained below.**
- Absences pre-approved by the UA Dean of Students (or Dean Designee) will be honored. See: https://deanofstudents.arizona.edu/absences.
- The UA policy regarding absences for any sincerely held religious belief, observance or practice will be accommodated where reasonable, http://policy.arizona.edu/human-resources/religious-accommodation-policy.

Syllabus

Required Text

- None

Required or Special Materials

- All required software will be available online at no cost to the student.
- However, students are expected to either have a laptop/desktop capable of handling homework and classwork.
- Mac, PC (Windows 10) or Linux.

Syllabus

Final Examination or Project

- No examinations, e.g. mid-term or final, are scheduled for this course.

Grading Scale and Policies

- homework exercises (50%)
- a term programming project (50%)
- **ungraded homework exercises too**
- Requests for incomplete (I) or withdrawal (W) must be made in accordance with University policies, which are available at http://catalog.arizona.edu/policy/grades-and-grading-system#incomplete and http://catalog.arizona.edu/policy/grades-and-grading-system#Withdrawal respectively.

Syllabus

Assignments and Examinations: Schedule/Due Dates

- All homeworks will be introduced **and reviewed** in class.
- Homework submissions by email to me only.
- Late homework will be not accepted since all homeworks will be solved/reviewed in class.
- Quick homeworks are normally due at midnight before the next class, and are generally assigned in class on a **Tuesday** and due **Wednesday** midnight (before **Thursday's** class).
- Homeworks not categorized as quick are normally assigned in class on a **Thursdays** and due the following **Monday** midnight (before next **Tuesday's** class). (Some longer homeworks may have an extended due date.)
- Students can expect a total of around 8-10 homeworks over the course.

Syllabus

Code of Academic Integrity

- You may discuss homework questions with anyone or anything.
- You may look things up on the web and use answers found therein; however, you must write it up yourself (in your own words/own code *etc*.).
- You must cite all (web) references, including ChatGPT, and your classmates (in the case of shared discussion).
- Students are encouraged to share views and discuss freely the principles and applications of course materials.
- However, graded work/exercises must be the product of independent effort unless otherwise instructed.
- Students are expected to adhere to the UA Code of Academic Integrity as described in the UA General Catalog. See: http://deanofstudents.arizona.edu/academic-integrity/students/academic-integrity.

Syllabus

UA Nondiscrimination and Anti-harassment Policy

- The University is committed to creating and maintaining an environment free of discrimination; see http://policy.arizona.edu/human-resources/nondiscrimination-and-anti-harassment-policy.

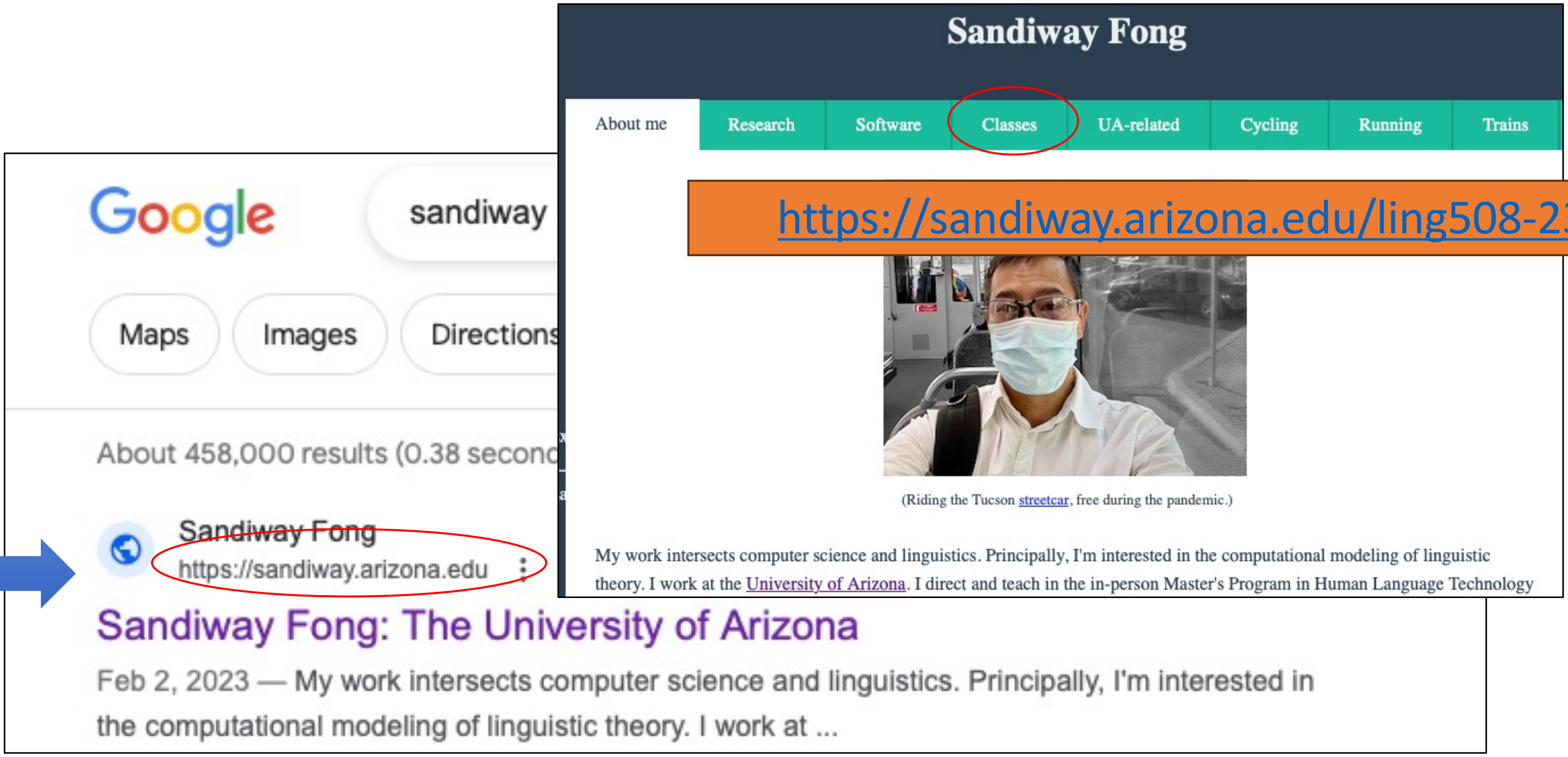
Subject to Change Statement

- Information contained in the course syllabus, other than the grade and absence policy, may be subject to change with advance notice, as deemed appropriate by the instructor.

Syllabus

- Questions?

Course website



https://sandiway.arizona.edu/ling508-23/

Sandiway Fong

(Riding the Tucson streetcar, free during the pandemic.)

My work intersects computer science and linguistics. Principally, I'm interested in the computational modeling of linguistic theory. I work at the University of Arizona. I direct and teach in the in-person Master's Program in Human Language Technology

About 458,000 results (0.38 second

1st hit

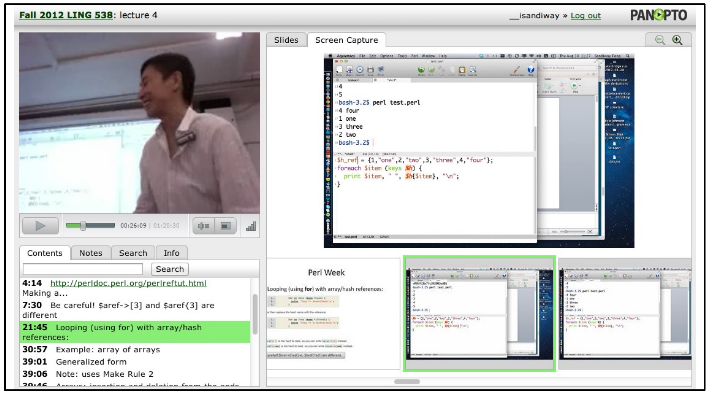
Sandiway Fong
https://sandiway.arizona.edu

Sandiway Fong: The University of Arizona

Feb 2, 2023 — My work intersects computer science and linguistics. Principally, I'm interested in the computational modeling of linguistic theory. I work at ...

Panopto

- Download lecture slides from my course homepage
  - https://sandiway.arizona.edu/ling508-23/
  - available from just before class time
    - (afterwards, look again for updates and corrections)
  - in .pptx (good for animations) and .pdf formats
- Lectures will be recorded using the Panopto system
  - accessible via the course webpage
  - sometimes crashes
  - (video, terminal screen, synchronized slides, keyword search)



```
4
5
bash-3.2$ perl test.perl
4 four
1 one
3 three
2 two
bash-3.2$
```

```
$h_ref = {1,"one",2,"two",3,"three",4,"four"};
foreach $item (keys %h) {
  print $item, " ", $h{$item}, "\n";
}
```

4:14 http://perldoc.perl.org/perlreftut.html Making a...

7:30 Be careful! $aref->[3] and $aref{3} are different

21:45 Looping (using for) with array/hash references:

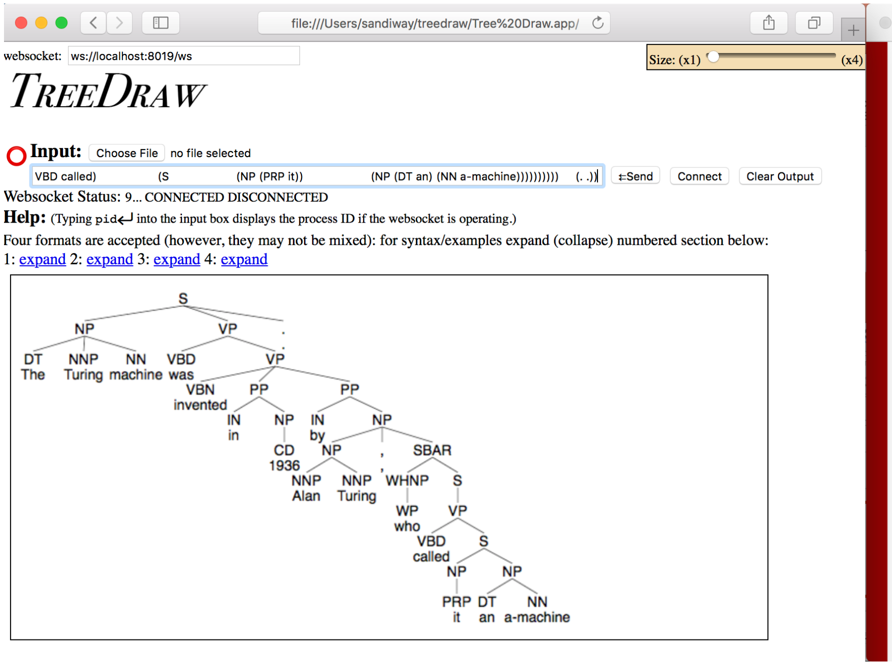
30:57 Example: array of arrays

39:01 Generalized form

39:06 Note: uses Make Rule 2

39:46 Arrays: insertion and deletion from the ends

Example



file:///Users/sandiway/treedraw/Tree%20Draw.app/

websocket: ws://localhost:8019/ws

Size: (x1) (x4)

TREEDRAW

○ **Input:** Choose File no file selected

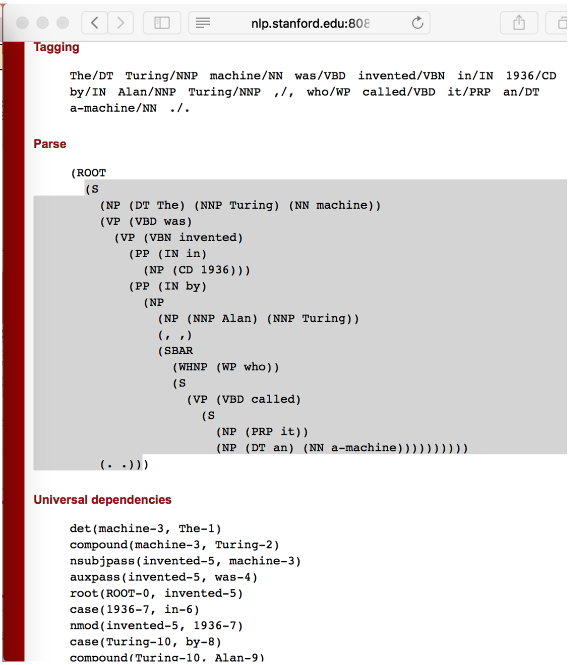
```
VBD called)          (S          (NP (PRP it))          (NP (DT an) (NN a-machine)))))))))     (. .))
```

⇇Send Connect Clear Output

Websocket Status: 9... CONNECTED DISCONNECTED

Help: (Typing `pid↵` into the input box displays the process ID if the websocket is operating.)

Four formats are accepted (however, they may not be mixed): for syntax/examples expand (collapse) numbered section below:
1: expand 2: expand 3: expand 4: expand



nlp.stanford.edu:808

Tagging

```
The/DT  Turing/NNP  machine/NN  was/VBD  invented/VBN  in/IN  1936/CD
by/IN  Alan/NNP  Turing/NNP  ,/,  who/WP  called/VBD  it/PRP  an/DT
a-machine/NN  ./.
```

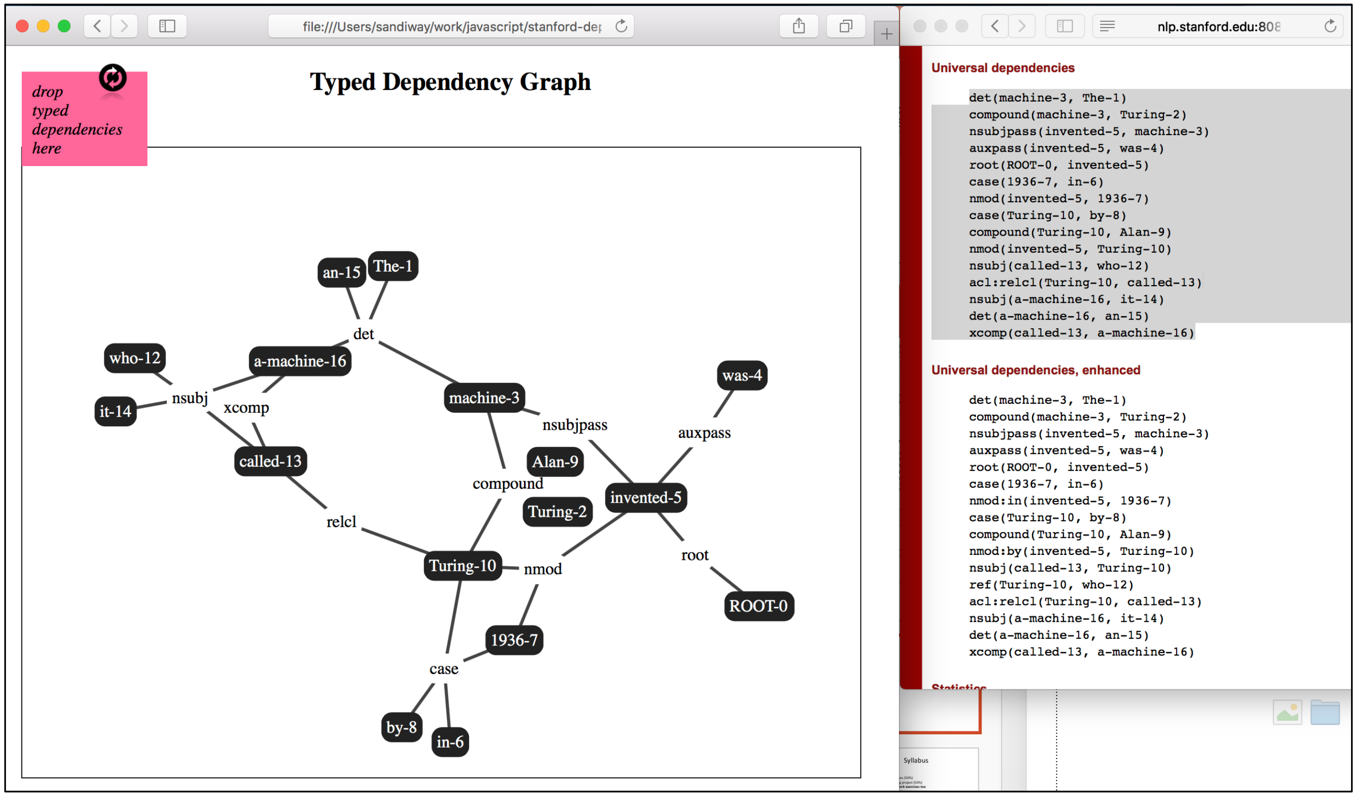
Parse

```
(ROOT
  (S
    (NP (DT The) (NNP Turing) (NN machine))
    (VP (VBD was)
      (VP (VBN invented)
        (PP (IN in)
          (NP (CD 1936)))
        (PP (IN by)
          (NP
            (NP (NNP Alan) (NNP Turing))
            (, ,)
            (SBAR
              (WHNP (WP who))
              (S
                (VP (VBD called)
                  (S
                    (NP (PRP it))
                    (NP (DT an) (NN a-machine))))))))))
    (. .)))
```

Universal dependencies

```
det(machine-3, The-1)
compound(machine-3, Turing-2)
nsubjpass(invented-5, machine-3)
auxpass(invented-5, was-4)
root(ROOT-0, invented-5)
case(1936-7, in-6)
nmod(invented-5, 1936-7)
case(Turing-10, by-8)
compound(Turing-10, Alan-9)
```

Example



Universal dependencies

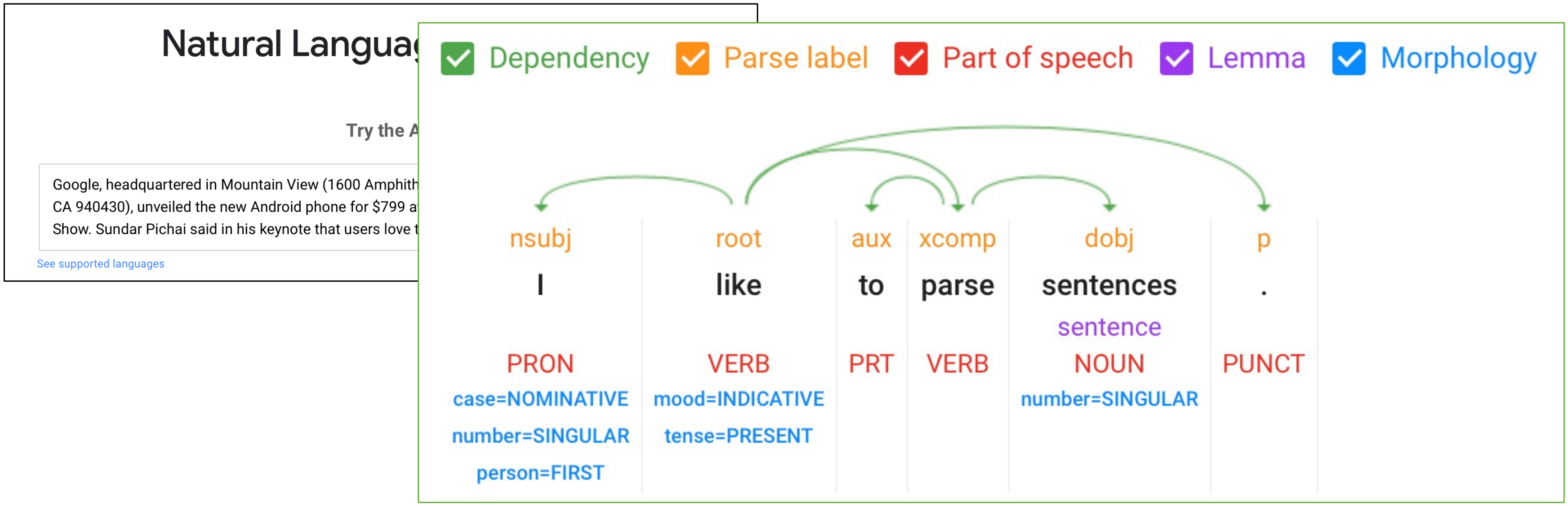
```
det(machine-3, The-1)
compound(machine-3, Turing-2)
nsubjpass(invented-5, machine-3)
auxpass(invented-5, was-4)
root(ROOT-0, invented-5)
case(1936-7, in-6)
nmod(invented-5, 1936-7)
case(Turing-10, by-8)
compound(Turing-10, Alan-9)
nmod(invented-5, Turing-10)
nsubj(called-13, who-12)
acl:relcl(Turing-10, called-13)
nsubj(a-machine-16, it-14)
det(a-machine-16, an-15)
xcomp(called-13, a-machine-16)
```

Universal dependencies, enhanced

```
det(machine-3, The-1)
compound(machine-3, Turing-2)
nsubjpass(invented-5, machine-3)
auxpass(invented-5, was-4)
root(ROOT-0, invented-5)
case(1936-7, in-6)
nmod:in(invented-5, 1936-7)
case(Turing-10, by-8)
compound(Turing-10, Alan-9)
nmod:by(invented-5, Turing-10)
nsubj(called-13, Turing-10)
ref(Turing-10, who-12)
acl:relcl(Turing-10, called-13)
nsubj(a-machine-16, it-14)
det(a-machine-16, an-15)
xcomp(called-13, a-machine-16)
```

Syntactic Parsing

- Google Natural Language
- https://cloud.google.com/natural-language/



Natural Languag

Try the A

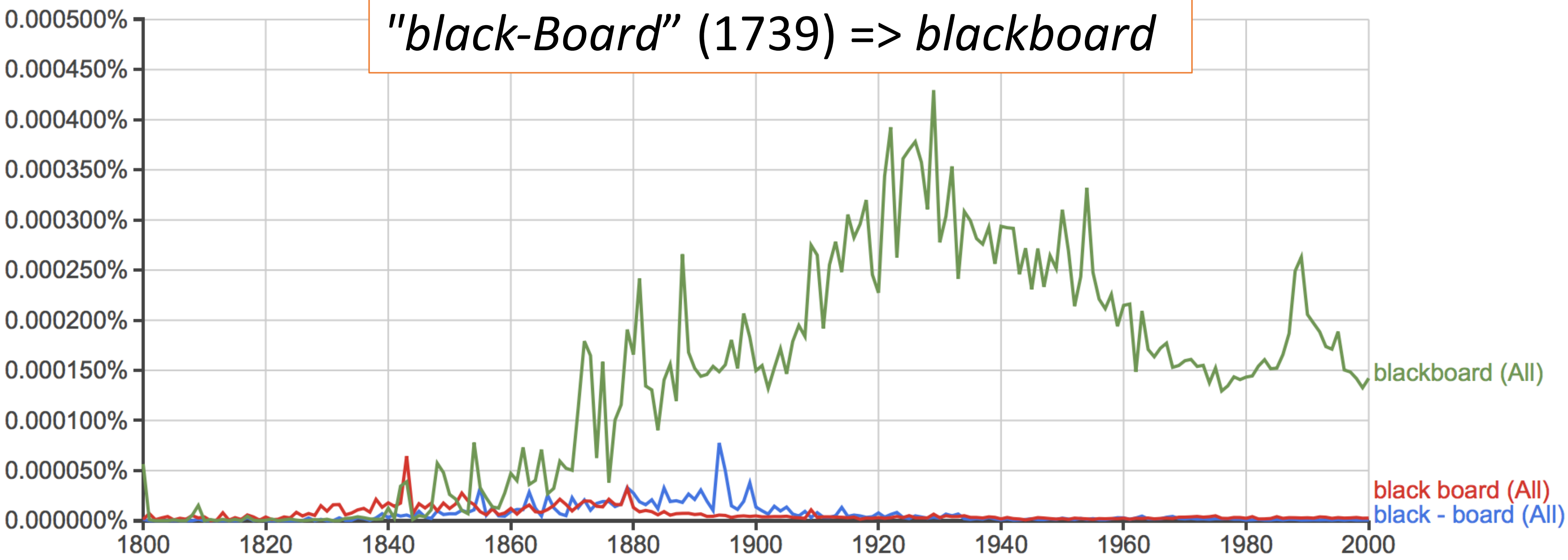
Google, headquartered in Mountain View (1600 Amphith
CA 940430), unveiled the new Android phone for $799 a
Show. Sundar Pichai said in his keynote that users love t

See supported languages

☑ Dependency ☑ Parse label ☑ Part of speech ☑ Lemma ☑ Morphology

nsubj	root	aux	xcomp	dobj	p
I	like	to	parse	sentences	.
				sentence	
PRON	VERB	PRT	VERB	NOUN	PUNCT
case=NOMINATIVE number=SINGULAR person=FIRST	mood=INDICATIVE tense=PRESENT			number=SINGULAR	

Google n-grams

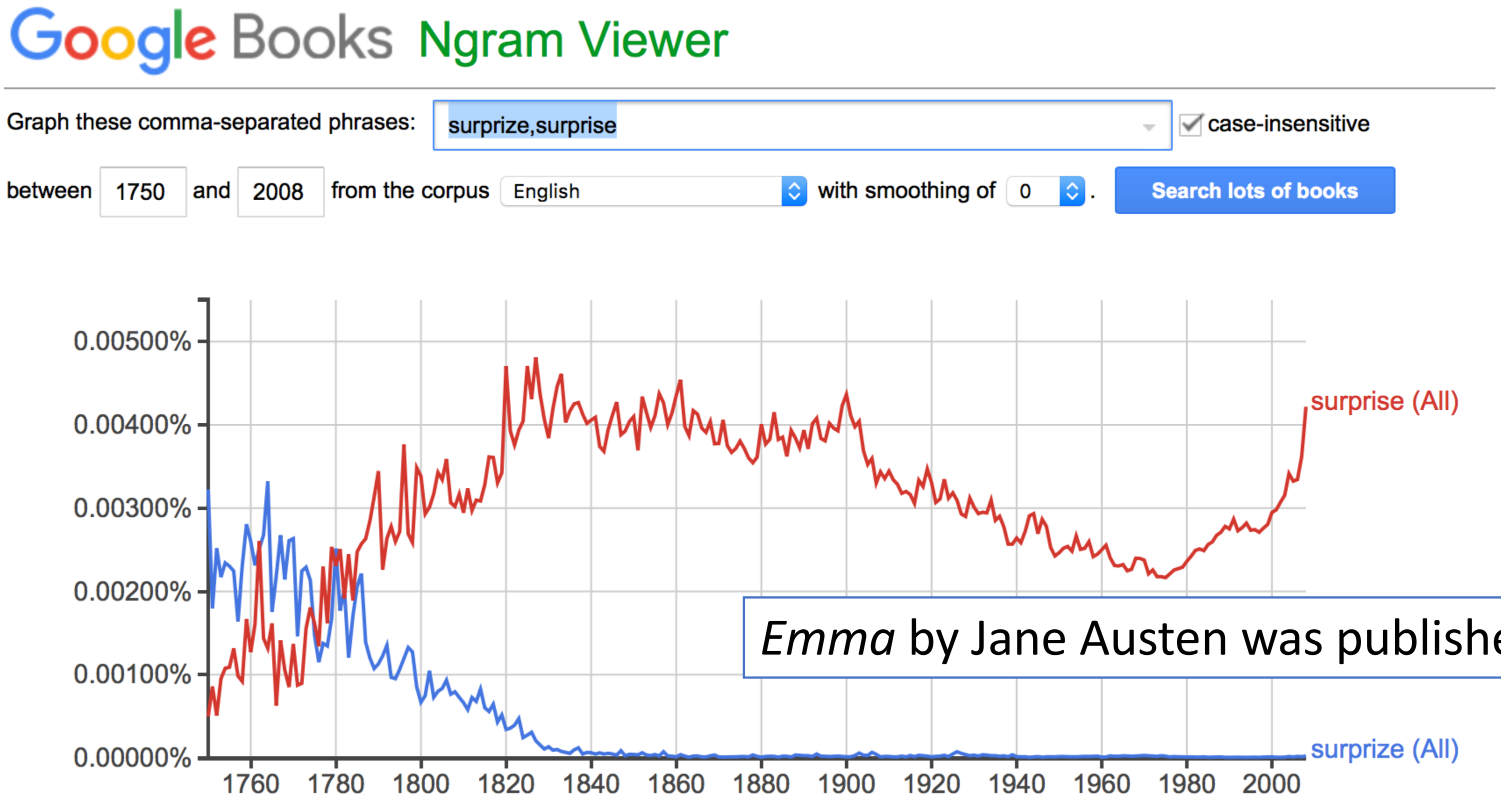


Staged word compound formation?

"black-Board" (1739) => *blackboard*

https://books.google.com/ngrams

Google: relative frequency of two spellings



Google Books Ngram Viewer

Graph these comma-separated phrases: surprize,surprise ☑ case-insensitive

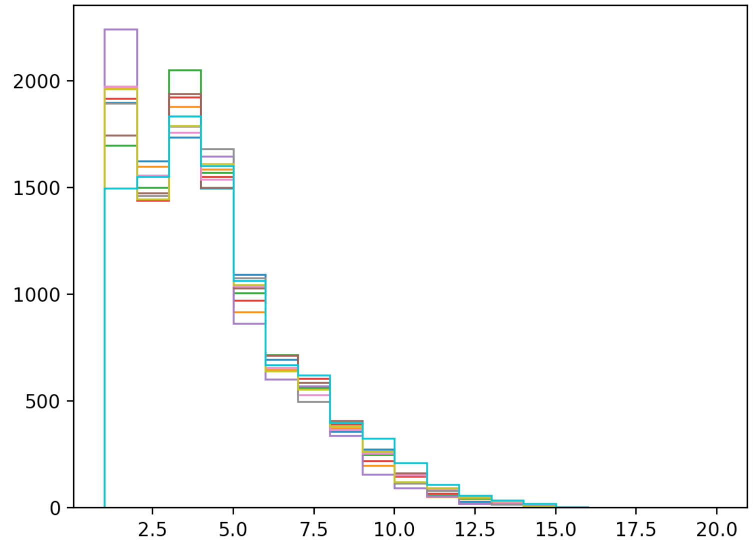
between 1750 and 2008 from the corpus English with smoothing of 0 .

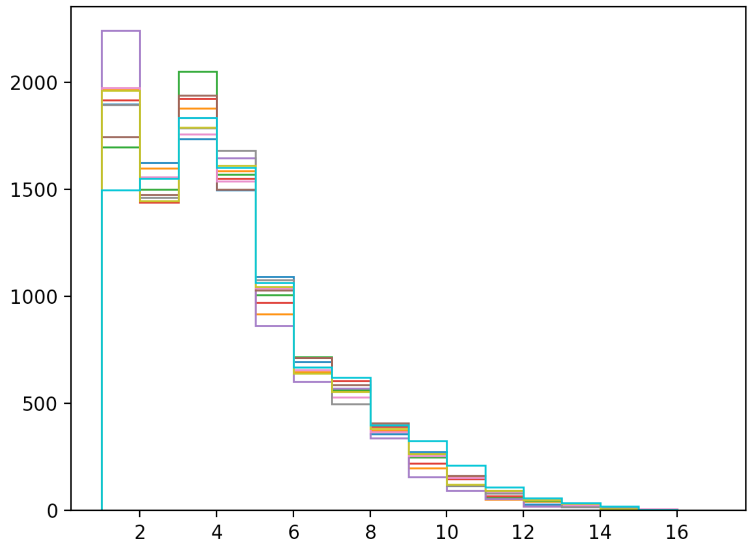
Emma by Jane Austen was published in 1815

Stylometry: compare word length distribution

```
len1s = [len1[i*10000:i*10000+10000] for i in range(10)]
for l in len1s:
    plt.hist(l, bins=np.arange(min(l),max(l)+1), histtype='step')
plt.show()
```

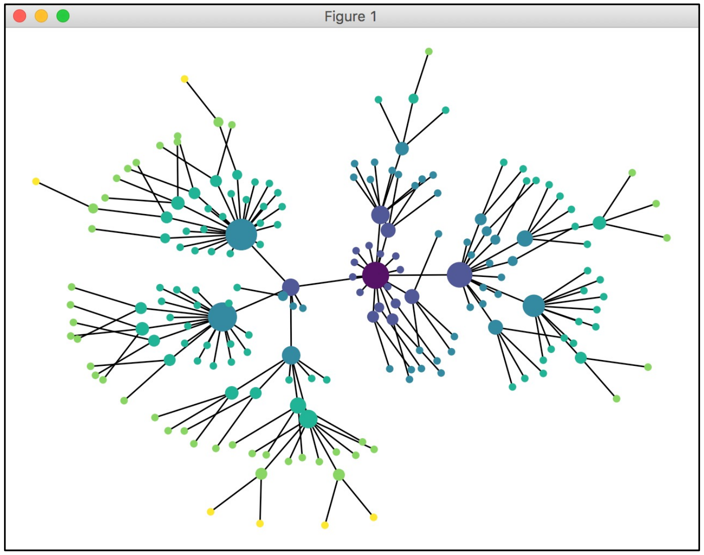
Forensic linguistics

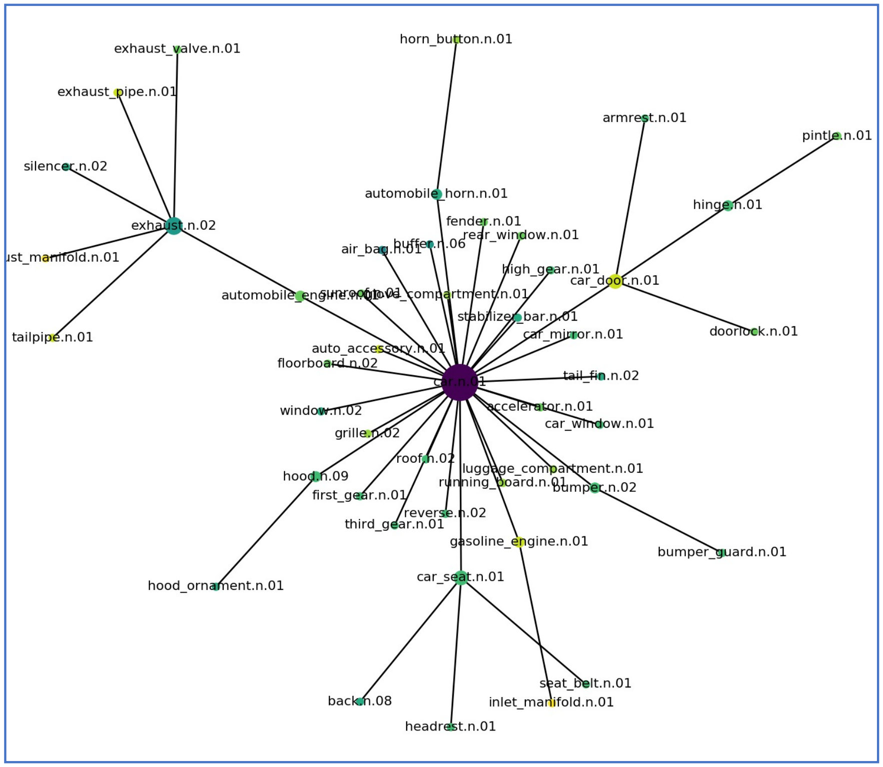




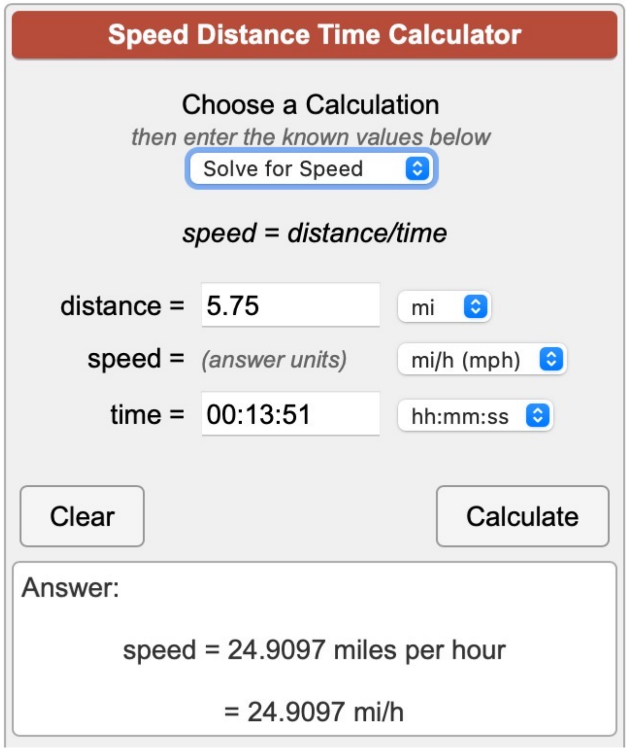
WordNet relations: parts of a car

```
from nltk.corpus import wordnet as wn
c = wn.synset('car.n.01')
g = graph(c, 'part_meronyms')
graph_draw(g)
```





Browser language: Javascript



Speed Distance Time Calculator

Choose a Calculation

then enter the known values below

Solve for Speed

speed = distance/time

distance = 5.75 mi

speed = *(answer units)* mi/h (mph)

time = 00:13:51 hh:mm:ss

Answer:

speed = 24.9097 miles per hour

= 24.9097 mi/h

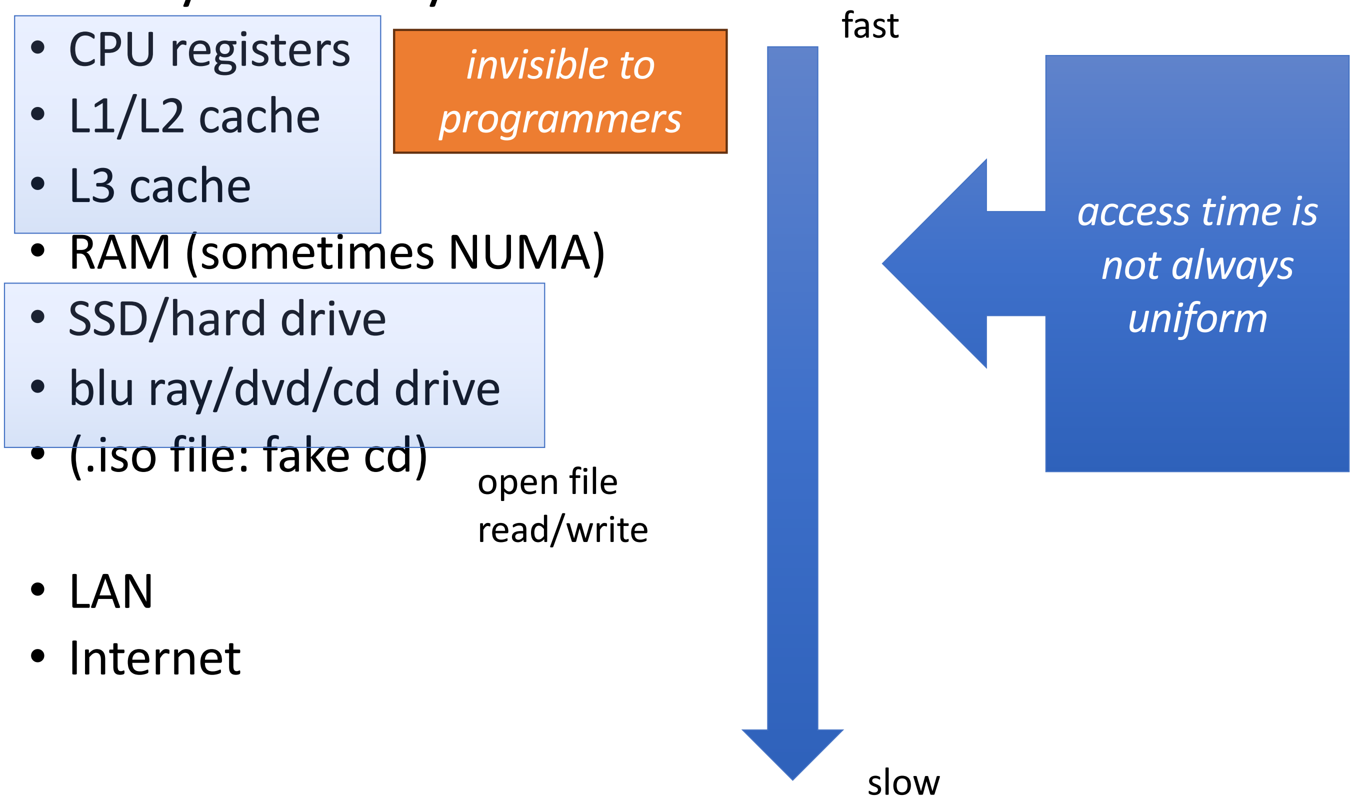
- Does this run in your browser?
- Or does your browser send a request to a webserver to compute the result and send it back to your browser (when you hit calculate)?
- We will show how both paradigms work in this class
  - you will get the opportunity to run a webserver on your laptop (Apache2).

Introduction

- Computers
    - Memory (several kinds)
        - Programs and data
    - CPU (Central Processing Unit)
        - Interprets machine instructions
    - I/O (Input/Output)
        - keyboard, mouse, touchpad, screen, touch sensitive screen, printer, usb port, etc.
        - bluetooth, usb, thunderbolt, ethernet, wifi, cellular …

Introduction

- Memory hierarchy



  - CPU registers
  - L1/L2 cache
  - L3 cache
  - RAM (sometimes NUMA)
  - SSD/hard drive
  - blu ray/dvd/cd drive
  - (.iso file: fake cd)
  - LAN
  - Internet

invisible to programmers

open file read/write

fast

slow

access time is not always uniform