LING 388: Computers and Language

Lecture 2
Administrivia

• Introducing your TA (Patricia Lee, pllee@email.arizona.edu)
• Have you installed Python (2 and 3)?
Today's Topics

• As mentioned in the syllabus:
  1. Introduce you to natural language processing (tools etc.)
  2. Introduce you to programming (Python)

• We will proceed along the two tracks simultaneously ...
Star Wars and Star Trek

From last time...
Ronald Reagan "Star Wars" Speech

https://www.youtube.com/watch?v=8phZrqb8QOY

- Top bigrams and trigrams:

<table>
<thead>
<tr>
<th>Bigram</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>('Soviet', 'Union')</td>
<td>16</td>
</tr>
<tr>
<td>('United', 'States')</td>
<td>8</td>
</tr>
<tr>
<td>('nuclear', 'weapons')</td>
<td>6</td>
</tr>
<tr>
<td>('defense', 'budget')</td>
<td>4</td>
</tr>
<tr>
<td>('conventional', 'forces')</td>
<td>4</td>
</tr>
<tr>
<td>('military', 'forces')</td>
<td>4</td>
</tr>
<tr>
<td>('nuclear', 'war')</td>
<td>4</td>
</tr>
<tr>
<td>('vital', 'interests')</td>
<td>4</td>
</tr>
<tr>
<td>('Federal', 'budget')</td>
<td>3</td>
</tr>
<tr>
<td>('years', 'ago')</td>
<td>3</td>
</tr>
<tr>
<td>('defense', 'spending')</td>
<td>3</td>
</tr>
<tr>
<td>('arms', 'control')</td>
<td>3</td>
</tr>
<tr>
<td>('ballistic', 'missiles')</td>
<td>3</td>
</tr>
<tr>
<td>('20', 'years')</td>
<td>3</td>
</tr>
<tr>
<td>('arms', 'reduction')</td>
<td>3</td>
</tr>
</tbody>
</table>

Bigram Star Wars was never actually mentioned

You will learn how to write a program to do this in this course ☞
Blackboards and Whiteboards

• **Whiteboards:**
  • "Dry-erase markers for whiteboards were invented in 1975."
  • "They became more common in classrooms during the 1990s due to concerns over health problems in children with dust allergies and the potential for chalk dust to damage computers."
  • "By the late 1990s, about 21% of American classrooms had converted from chalkboards to whiteboards."

• **Blackboards:**
  • "The term ‘blackboard’ is attested in English from the mid-eighteenth century."
  • OED cites: “*with Chalk on a black-Board*” (1739)
  • "The first attested use of chalk on blackboard in the United States dates to September 21, 1801, in a lecture course in mathematics given by George Baron."

(Source: Wikipedia)
Blackboards and Whiteboards

Staged word compound formation?
"black-Board" (1739) => blackboard
Blackboards and Whiteboards

whiteboard / white board / white-board
Blackboards and Whiteboards
Toothpaste vs tooth paste vs tooth-paste

also tooth-paste, 1832, from tooth + paste (n.).
Earlier substances were tooth-powder (1540s); tooth-soap (c. 1600).
database
Homework 2

• Due next Monday (midnight)
• Email homework answers to pllee@email.arizona.edu
• Subject: LING 388 Homework 2
• Submit one PDF file. Put your name at the top.
Question 1

• Look up **egghead** in all possible forms on Google n-grams
  • *egghead, egg-head, egg head* (check: *case insensitive*)
  • *x-axis* (use dates: e.g. 1800 to 2008)

• Can you explain the biggest bump on the y-axis (frequency) for the closed form: *egghead*?
Homework 2

Question 2
• Look up the closed form *happenstance* in Google n-grams
• Is it a relatively new word? Explain the chart

Question 3
• Compare *happenstance* and *happenchance*
• Can you speculate on possible reasons why one is more frequent than the other?
Python: Numbers

It's time to get started...

• Data structures are the one of the two fundamental topics we have to tackle (the other will be control structures)
• Let's start with numbers
• (Python book chapter 3 (pgs.51–69): read it!)
>>> type(3)
<type 'int'>

>>> type(3.14)
<type 'float'>

>>> type(3.0)
<type 'float'>

>>> myInt = -32
>>> type(myInt)
<type 'int'>

>>> myFloat = 32.0
>>> type(myFloat)
<type 'float'>

<table>
<thead>
<tr>
<th>operator</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>addition</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
</tr>
<tr>
<td>**</td>
<td>exponentiation</td>
</tr>
<tr>
<td>%</td>
<td>remainder</td>
</tr>
<tr>
<td>abs()</td>
<td>absolute value</td>
</tr>
</tbody>
</table>

Table 3.1: Python built-in numeric operations.
import math
math.pi

>>> from math import pi, sin
>>> sin(pi/2)
1.0

<table>
<thead>
<tr>
<th>Python</th>
<th>Mathematics</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>pi</td>
<td>$\pi$</td>
<td>An approximation of pi.</td>
</tr>
<tr>
<td>e</td>
<td>$e$</td>
<td>An approximation of $e$.</td>
</tr>
<tr>
<td>sin(x)</td>
<td>$\sin x$</td>
<td>The sine of x.</td>
</tr>
<tr>
<td>cos(x)</td>
<td>$\cos x$</td>
<td>The cosine of x.</td>
</tr>
<tr>
<td>tan(x)</td>
<td>$\tan x$</td>
<td>The tangent of x.</td>
</tr>
<tr>
<td>asin(x)</td>
<td>$\arcsin x$</td>
<td>The inverse of sine x.</td>
</tr>
<tr>
<td>acos(x)</td>
<td>$\arccos x$</td>
<td>The inverse of cosine x.</td>
</tr>
<tr>
<td>atan(x)</td>
<td>$\arctan x$</td>
<td>The inverse of tangent x.</td>
</tr>
<tr>
<td>log(x)</td>
<td>$\ln x$</td>
<td>The natural (base $e$) logarithm of x</td>
</tr>
<tr>
<td>log10(x)</td>
<td>$\log_{10} x$</td>
<td>The common (base 10) logarithm of x.</td>
</tr>
<tr>
<td>exp(x)</td>
<td>$e^x$</td>
<td>The exponential of x.</td>
</tr>
<tr>
<td>ceil(x)</td>
<td>$\lceil x \rceil$</td>
<td>The smallest whole number $\geq x$</td>
</tr>
<tr>
<td>floor(x)</td>
<td>$\lfloor x \rfloor$</td>
<td>The largest whole number $\leq x$</td>
</tr>
</tbody>
</table>

Table 3.2: Some math library functions.
Introduction

• Memory Representation
  • binary: zeros and ones (1 bit)
  • organized into bytes (8 bits)
    • memory is byte-addressable
  • word (32 bits)
    • e.g. integer
    • (64 bits: floating point number)
  • big-endian/little-endian
    • most significant byte first or least significant byte
    • communication ...

0

array a[23]

addressable Memory (RAM)

FFFFFFFF

your Intel and ARM CPUs
Introduction

• A typical notebook computer
  • Example: a 2013 Macbook Air
  • CPU: Core i5-4250U
    • 1.3 billion transistors
    • built-in GPU
    • TDP: 15W (1.3 GHz)
    • Dual core (Turbo: 2.6 GHz)
    • Hyper-Threaded (4 logical CPUs, 2 physical)
    • 64 bit
    • 64 KB (32 KB Instruction + 32 KB Data) L1 cache
    • 256 KB L2 cache per core
    • 12MB L3 cache shared
    • 16GB max RAM

Increased address space and 64-bit registers
Introduction

4th Generation Intel® Core™ Processor Die Map
22nm Tri-Gate 3-D Transistors

Quad core die shown above  Transistor count: 1.4 Billion  Die size: 177mm²
A 4 core machine: 8 virtual

anandtech.com
Introduction

• Machine Language
  • A CPU understands only one language: machine language
    • all other languages must be translated into machine language
  • Primitive instructions include:
    • MOV
    • PUSH
    • POP
    • ADD / SUB
    • INC / DEC
    • IMUL / IDIV
    • AND / OR / XOR / NOT
    • NEG
    • SHL / SHR
    • JMP
    • CMP
    • JE / JNE / JZ / JG / JGE / JL / JLE
    • CALL / RET

Assembly Language: (this notation) by definition, nothing built on it is more powerful

http://www.cs.virginia.edu/~evans/cs216/guides/x86.html
Introduction

• Not all the machine instructions are conceptually necessary
  • many provided for speed/efficiency

• Theoretical Computer Science
  • All mechanical computation can be carried out using a **TURING MACHINE**
  • Finite state table + (infinite) tape
  • Tape instructions:
    • at the tape head: Erase, Write, Move (Left/Right/NoMove)
  • Finite state table:
    • Current state x Tape symbol --> new state x New Tape symbol x Move
Introduction

• Storage:
  • based on digital logic
  • binary (base 2) – everything is a power of 2
  • Byte: 8 bits
    • 01011011
    • = $2^6+2^4+2^3+2^1+2^0$
    • = $64 + 16 + 8 + 2 + 1$
    • = 91 (in decimal)
  • Hexadecimal (base 16)
    • 0-9,A,B,C,D,E,F (need 4 bits)
    • 5B (= 1 byte)
    • = $5*16^1 + 11$
    • = 80 + 11
    • = 91
Introduction: data types

• Integers
  • In one byte (= 8 bits), what’s the largest and smallest number, we can represent?
  • **Answer:** -128 .. 0 .. 127
  • Why? -2^{8-1} .. 0 .. 2^{8-1} – 1
  • 00000000 = 0
  • 01111111 = 127
  • 10000000 = -128
  • 11111111 = -1

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2^8 – 1 = 255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2^7 – 1 = 127</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>