The final

- At officially scheduled time next Tuesday at 8 AM
- Covers the material not covered on the midterm
- Not cumulative
- Multiple Choice. Bring a PINK SCANTRON, PENCILS
- Will be longer than the midterm but I don’t think for most people it will take all three hours.
- These slides select things you should know, should have learned the second half of the class.

Asimov’s Laws: A robot...

1. Must not harm a human being through action or inaction
2. Must obey all human orders, except in conflict with Rule 1
3. Must protect its own existence, except in conflict with Rule 1 and 2

How do we prevent robots from injuring themselves or failing?

- Static Environment
- Sensors
- Coding

Often robot failure is the fault of design requirements.

How would we wish to apply rule 2?

- Gestures
- Common Speech
- Robot recognizing our patterns
- Robots learning

Learning: Which robots use learning?

- Leonardo (MIT)
- BigDog (Boston Dynamics)
What kinds of things could Baxter do?

Baxter $22,000

SWARM ROBOTICS:

what might they be good for? search and rescue.

Originals:
- 2003-2008
- MIT/James McLurkin

Parallel:
- 2005-2012
- Universite Libre de Bruseels/Marco Dorigo

Cheap ($250)
- 2009-2012
- Rice/James McLurkin

what is ergonomics?

- a. the study of people’s efficiency in their working environment
- b. the mental action or process of acquiring knowledge and understanding through thought
- c. the mental action or process of acquiring knowledge and understanding through experience
- d. the mental action or process of acquiring knowledge and understanding through the senses

The following picture was used to show:

A. Children with Autism have a very poor ability to remember.
B. Children with Autism prefer to find images on the web than to draw them themselves.
C. Children with Autism have enormous difficulty even talking about the pictures they make.
D. Children who normally are unable to draw well can produce creative artwork when given tools that are adapted to their limitations and strengths.
E. All of the above.

One fact that Prof. Kurniawan pointed out was that:

A. The best you can do with human computer interface design is to design systems that work well for males between 14 and 22.
B. Kaleidoscopes are a better computational art technology.
C. Black and White text on the screen is actually very bad for people with Autism.
D. All of the above.
E. None of the above.
Prof. Kurniawan also pointed out that:

A. Everyone can be considered disabled at some point in their life
B. Assistive technologies often become useful to the mainstream population
C. HCI and AT are complementary research perspectives
D. All of the above
E. None of the above

- My user group ranges from people with disabilities, the aging population, children, those with low literacy/socio economic status and those from third world countries

Do Blind people take photos for same reasons as sighted people?

What is Prof. Kurniawan’s system for Blind Photography called?

What does it let you do?

Guest Lecture: Computational Cinematics
Prof. Arnav Jhala
Computer Cinematography involves:

- Question with various list elements or not.
- lighting,
- camera control,
- staging,
- writing,
- editing,
- viewer experience,
- character.

A game that uses the same visual imagery as Citizen Kane is:

"The mirror causes the image to repeat infinitely. Deep focus is used to enhance the repetition, which adds to Kane's loneliness as an old man and to his isolation."

-James West, AFS

A game with the same camera continuous action shooting action as Rope is:

- MazeBall
- Panorama
- Portal

I asked myself whether it was technically possible to film it in the same way [real-time]. The only way to achieve that, I found, would be to handle the shooting in the same continuous action, with no break in the telling of a story that begins at seven-thirty and ends at nine-fifteen.

-Hitchcock (Interview with Truffaut)

What can Gamersourcing be used for?

Challenge #1: Learning Aesthetic Preferences

Which composition is better?

How would you do it automatically?

What is the rule that is used

(a)  
(b)  
(c)
Data Collection

- Three methods
  - UCSC students
  - Facebook
  - Amazon’s Mechanical Turk
- Two studies
  - Pairwise 4-AFC* (A is better, B is better, Both good, Both bad)
  - Absolute (1-6: bad to good)

The Panorama game

- Aesthetic features for evaluation of photographs that are represented in the Panorama game are:
  - A. Rule of Thirds
  - B. Symmetry
  - C. Number of objects in the scene
  - D. All of the above

Features

- Ranking scheme for collecting data on pairwise preferences that was used in the MazeBall game
  - (a) Likert scale (1-5)
  - (b) 2-AFC (alternative forced choice -- A or B)
  - (c) 4-AFC (A, B, Both, Neither)
  - (d) None of the above

Player Modeling

- Blood Volume Pulse (BVP) → Heart Rate (HR)
- Skin Conductance (SC)

How to make a game more challenging/stressful?

- A. Increase number of prizes
- B. Bring the camera angle closer to the character
- C. Spread any enemies or obstacles out more evenly
- D. Use the rule of thirds
Guest Lecture: ME
Expressive Generation for Interactive Stories

Natural Language and Dialogue Systems Lab

Procedural Language Generation: A Key Technology

- Wide range of generation parameters
- Different methods for creating models that control the parameters
  - Dynamic Real-Time Adaptation
  - Trainable: Machine Learning Techniques
  - Individual Personalization

PERSONAGE Generator: BIG FIVE Theory

- Conscientiousness: Dutiful vs. impulsive
- Emotional stability: Calm vs. anxious
- Openness to experience: Imaginative vs. conventional
- Agreeableness: Kind vs. unfriendly
- Extraversion: Sociable, assertive vs. quiet

Ist method: Rule-Based Extraversion Generation

- Use correlations in literature to set parameters
- Significant perceptual differences \( p < .01 \)
- As binary classification, 90% accuracy

Annie Hall: Getting a lift

Scene from Annie Hall: Lobby of Sports Club
ALVY: Uh ... you-you wanna hit?
ANNIE: Turning and aiming her thumb over her shoulder
Oh, why uh ... y-y-you gotta car?
ALVY: No, um ... I was gonna take a cab.
ANNIE: Laughting: Oh, no, I have a car.
ALVY: You have a car? Annie smiles, hands folded in front of her
So ... Clears his throat: I don't understand why ... if you have a car, so then-then why did you say “Do you have a car?” ... like you wanted a lift!

Method

1. Collect movie scripts from IMDB
2. Extract utterances for each character
3. Select leading roles (dialogue > 60 turns)
4. Generate features reflecting linguistic behaviors
Has anyone ever shown that it matters?

- People make attributions beyond social level: task competence
- Personality matching in a robotic exercise coach increased the time that stroke victims spent on their medically recommended exercises (Tapos & Mazaric 2008)
- Tutoring oriented to the student’s ‘face needs’ improved learning in training and tutoring (Porayska-Pomsta & Melish 2004; Wang et al., 2004)

What are you supposed to learn?

- HTML let’s you programmatically indicate how a particular content should be displayed.
- It can be served up by any HTTP server anywhere in the world.
- Typically uses UTF-8 encoding to guarantee being able to be shown

My Web page -- Metadata

- Title
- Font names
- Table Structure
- Where to put line breaks

HTML and the Web

- The Web uses http:// protocol
- Its asking for a Web page, which usually means a page expressed in hyper-text markup language, or HTML
  - Hyper-text refers to text containing links that allow you to leave the linear stream of text, see something else, and return to the place you left
  - Markup language is a notation to describe how a published document is supposed to look: what kinds of fonts, text color, headings, images, etc.
Basics of HTML #1

- Rule 1: Content is given directly; anything that is not content is given inside of tags.
- Rule 2: Tags made of `<` and `>` and used this way:
  
  `<p style="color:red">This is paragraph.</p>`
  
  It produces: This is paragraph.

- Rule 3: Tags must be paired or “self terminated.”

Basics of HTML #2

- Rule 4: An HTML file has this structure:
  
  `<html>
  <head><title>Name of Page</title></head>
  <body>
  Actual HTML page description goes here
  </body>
  </html>`

- Rule 5: Tags must be properly nested
- Rule 6: White space is mostly ignored
- Rule 7: Attributes (style="color:red") preceded by space, name not quoted, value quoted

Basics of HTML #3

- To put in an image (.gif, .jpg, .png), use 1 tag
  
  `<img src="cooking-ewan-isabel.jpg" alt="Kids Cooking"/>`

- To put in a link, use 2 tags
  
  `<a href="http://users.soe.ucsc.edu/~maw">Prof. Walker's</a>`

Example: myfirst.html

```html
<html>
  <head>
    <title>My First Page</title>
  </head>
  <body>
    <p>Here we go to eat it! </p>
    <h1>Make a Cake</h1>
    <img src="cooking-ewan-isabel.jpg" alt="Kids Cooking"
     style="width=300"/>
    <p>Here we go to eat it! </p>
  </body>
</html>
```

In Processing: File Export

- Makes an index.html file. You browse it!

Important Processing Concepts
Processing Concepts you NEED TO KNOW!

• Variables and Declarations
• Assignments
• Expressions
• Tests or If-statements
• NEWEST STUFF.
• Repetition (looping) or For-statements
• Functions and Functional Abstraction

For loops (Repetition)

• Repeating commands is powerful:
  • Lightbot 2.0 used recursion, a function calling itself
  • Symbolic Lightbot prefixed a number, 2:Step
• Processing uses a for loop:

```java
void setup() {
  size(500,200);
  background(0);
  noStroke();
  smooth();
  fill(255);
  for (int i = 0; i < 16; i++) {
    ellipse(100 + 25*i, 100, 15, 15);
  }
}
```

Repetition, Another Picture

• Or how about a bullseye?

```java
for (int i = 0; i < 5; i++) {
  ellipse(100 + 25*i, 100, 15, 15);
}
```

• Note the loop variable must be declared ... but could do it in loop itself like we did for pacman pills:

```java
for (int i = 0; i < 5; i++) {
  ellipse(100 + 25*i, 100, 15, 15);
}
```

Functional Abstraction Reduces Complexity

Layering: Building Functions out of Functions

Example: Abstraction in Everyday life

Abstraction

• Abstraction is a way to understand and solve problems
• As a general definition, abstraction eliminates details to focus on essential properties
• Functional abstraction gives a name to a series of operations that perform a coherent (and to us meaningful) activity; the name is the instruction, the series of operations are the actions to implement it
Functions In Processing: Result

- Functions that do something, but do not return a value, have `void` as their `return type`.
- Functions that return a value must say its type.

```java
void draw_a_box (int x_pos, int y_pos) {
  rect(x_pos, y_pos, 20, 20);
}
color pink () {
  return color(255, 200, 200);
}
```

Homework 10: Functions.

- Make sure you can reproduce what you did during final (e.g. be sure to understand the solution code)
- void cell(int x, int y, int s, color tint)
- void triple(int x, int y, int s)
- Also functions for:
  - block()
  - row()
  - cellarray()

Functional Abstraction: Layers of Functions

```
three_digit
  set
  digit one two three four five six seven eight nine zero
  leftup leftleft leftleftleft leftupleft upupupup rightup rightupright rightuprightright
  hexa rexa
  triangle rect triangle
```

Using Functions

```
void cell (int x, int y, int s, color tint) {
  rect(x, y, s, tint);
}
```

Arguments Become Parameters

- Notice that if the DEFINITION has n parameters, the CALL needs n arguments.
- The parameters and arguments correspond.

```
void draw ( ) {
  // (ii) (255);
  hexa(20, 40);
  hexa(40, 20);
  hexa(40, 40);
}
void hexa (int show, int show2) {
  rect(show, ybase+18, 28, 40);
  triangle(show, ybase+18, xbase+28, ybase+18, xbase+18, ybase);
  triangle(show, ybase+18, xbase+28, ybase+18, xbase+28, ybase+40);
}
```

Parameters

- Parameters are automatically declared (and initialized) on a call, and remain in existence as long as the function remains unfinished.
- When the function ends, the parameters vanish, only to be recreated on the next call.
- It is wise to choose parameter names that help you remember exactly what they mean.
  - colorFlag (Chao's code),
  - dir (for direction)
**Example: Clock Timer. In sample codes.**

- Draw digital timer elements
- Assemble elements into digits
- Light digit segments to create numbers
- Select number based on a digit

---

**Functional Abstraction: Layers of Functions**

```plaintext
three_digit
   set
   one  two three four five six seven eight nine zero

leftupper  leftlower  topupper  tophoriz  bottomlower  bottomhoriz

hexa  rexa

triangle, rect, triangle
```

---

**Using Functions**

- Once defined, functions can be called repeatedly ...
  it's the point of writing them!

```java
void setup() {
  size(200, 100);
  background(0);
  noLoop();
}

void draw() {
  void digit(float xbase, float ybase) {
    hexa(xbase, ybase+10); //left upper
    hexa(xbase, ybase+70); //left lower
    rexa(xbase+10, ybase); //mid horizontal
    rexa(xbase+10, ybase+60); //top horizontal
    rexa(xbase+10, ybase+120); //bot horizontal
    hexa(xbase+60, ybase+10); //right upper
    hexa(xbase+60, ybase+70); //right lower
  }
  Function  Cat
           hexa()  left upper
           rexa()  left lower
           bothoriz()  mid horizontal
           tophoriz()  top horizontal
           rightlower()  bot horizontal

  digit(90, 180);
}
```

---

**Clicker: I have hexa() and rexa(), now what?**

- A. I can make fun little shapes
- B. It obvious how I can make digits given these two shapes
- C. I have to define at least two more functions before I can make digits
- D. I think I can make digits but I just can't see how.

---

**Clicker: What does digit help you do?**

- A. You can re-use it over and over again
- B. It tells you what the coordinates are for the various bits of the digit
- C. If you pass the right arguments for xbase and ybase to digit you can make every digit
- D. A & B
- E. None of the above

---

**Use hexa, rexa to make a Digit**

```java
x = 123;

digit(90, 180);
```

---

**Clicker: What does digit help you do?**

- A. You can re-use it over and over again
- B. It tells you what the xbase and ybase to digit you can make every digit
- C. If you pass the right arguments for xbase and ybase to digit you can make every digit
- D. A & B
- E. None of the above
Let There Be Light (and Dark)

- Define the illumination of the digit
- `hexa_rexa_makeadigit_addlight.pde`
- Must declare two color variables, initialize to proper colors, use them in `fill()`, and then check them
  ```
  color dark, light;
  size(280, 180);
  background(light);
  
  void init() {
    light = color(255, 155, 0);
    dark = color(64, 64, 64);
    fill(dark);
    digit(0, 0);
    digit(10, 10);
  }
  ```

Clicker: What does the lighted digit help you do?

A. You can re-use it over and over again
B. It tells you what the coordinates are for the various bits of the digit
C. If you pass the right arguments for `xbase` and `ybase` to `digit` you can make every digit
D. A & B
E. None of the above

```
void digit(float xbase, float ybase) {
  // top left upper
  hexa(xbase, ybase+10);  // top left upper
  hexa(xbase, ybase+70);  // top left lower
  hexa(xbase+10, ybase);  // top horizontal
  hexa(xbase+10, ybase+30);  // top horizontal
  hexa(xbase+10, ybase+120);  // top horizontal
  // right lower
  hexa(xbase+60, ybase+70);  // right lower
  } // end digit
```

Clicker: What does the lighted digit help you do?

A. You can re-use it over and over again
B. It tells you what the coordinates are for the various bits of the digit
C. If you pass the right arguments for `xbase` and `ybase` to `digit` you can make every digit
D. A & B
E. None of the above

```
void digit(float xbase, float ybase) {
  // top left upper
  hexa(xbase, ybase+10);  // top left upper
  hexa(xbase, ybase+70);  // top left lower
  hexa(xbase+10, ybase);  // top horizontal
  hexa(xbase+10, ybase+30);  // top horizontal
  hexa(xbase+10, ybase+120);  // top horizontal
  // right lower
  hexa(xbase+60, ybase+70);  // right lower
  } // end digit
```

Clicker: What is time doing?

A. Nothing. It has a bug in it.
B. It is counting to infinity.
C. It is counting to 999 then starting at zero again
D. It counts up to 1000 then starts at 1.

```
void four(float xbase, float ybase) {
  hexa(xbase, ybase);  // left upper
  hexa(xbase+10, ybase);  // left lower
  rexa(xbase+10, ybase+10);  // mid horizontal
  hexa(xbase+60, ybase+10);  // right upper
  hexa(xbase+60, ybase+70);  // right lower
  } // end four
```

Clicker: this code for writing the digit 4 is ...

A. Perfect just as it is
B. Has a bug for right upper
C. Has a bug for mid horizontal
D. Actually makes a five

```
void four(float xbase, float ybase) {
  hexa(xbase, ybase);  // left upper
  hexa(xbase+10, ybase);  // left lower
  rexa(xbase+10, ybase+10);  // mid horizontal
  hexa(xbase+60, ybase+10);  // right upper
  hexa(xbase+60, ybase+70);  // right lower
  } // end four
```

Clicker: why would you want Count_in_lights_three_level?

A. I wouldn’t want it, it just makes the code even more confusing
B. Adding numbers to `ybase` and `xbase` for rexa and hexa can cause silly errors.
C. It makes the code more self documenting.
D. B & C

```
void leftupper(float xbase, float ybase) {
  hexa(xbase, ybase+10);  // left upper
  } // end leftupper
```

```
void leftlower(float xbase, float ybase) {
  hexa(xbase, ybase+70);  // left lower
  } // end leftlower
```

Clicker: why would you want Count_in_lights_three_level?

A. I wouldn’t want it, it just makes the code even more confusing
B. Adding numbers to `ybase` and `xbase` for rexa and hexa can cause silly errors.
C. It makes the code more self documenting.
D. B & C

```
void leftupper(float xbase, float ybase) {
  hexa(xbase, ybase+10);  // left upper
  } // end leftupper
```

```
void leftlower(float xbase, float ybase) {
  hexa(xbase, ybase+70);  // left lower
  } // end leftlower
```
Do function names matter?

Web Search

Needles in the Haystack: Google and Other Brokers in the Bits Bazaar

Blown To Bits Chapter 4

Important questions: What you should learn

- How can a search engine respond so fast?
- Does it find every relevant link?
- How does a search engine decide what gets listed first?
- If you try another search engine will you get the same result? If so, which is right? Which is better? Which is more authoritative?
- Are sponsored links better than "organic" links?
- Is the advertising necessary?
- What is the role of government? What should it be?

Web search: It Matters How It Works

1. Gather information.
2. Keep copies.
3. Build an index.
4. Understand the query.
5. Determine the relevance of each possible result to the query.
6. Determine the ranking of the relevant results.
7. Present the results.

Search Engines

No one controls what's published on the WWW ... it is totally decentralized

To find out, search engines crawl Web

- Two parts
  - Crawler visits Web pages building an index of the content (stored in a database)
  - Query processor checks user requests against the index, reports on known pages [You use this!]

Only a fraction of the Web's content is crawled

How many hosts are out there?
1. Gather Information
- Spiders or web crawlers wander the web building indices
- Estimates range from .02% to 3% of information is indexed
- How often does a page get visited?
  - some frequently (daily whitehouse.gov), others rarely
  - Crawler keeps track of which pages change frequently

2. Keep Copies
- Spider downloads the page as part of the “visit” in order to create the index.
- Search engine may “cache” the copy.
- Is this legal? What about copyright?
- But wait, browsing requires copying as well.

NEW YORK — Google and publishers told a US judge Thursday they are close to settling a lawsuit over the Internet giant’s controversial book-scanning project…”

3. Build an Index
- list of terms and for each term a list of where it appeared
- more than just the terms
  - terms in bigger font might be more important
  - terms in the title might be more important
- must be very fast to lookup
- could be millions of entries (not just words, but names, special numbers, etc.) requiring Gigabytes of memory
- must fit in the computer’s memory (see next slide)
Google indexes what percent of the pages in the world?
A. 100% It indexes everything.
B. About 50% on average but some days it’s 100% and some days it’s 30%
C. Less than 5%
D. Best estimates put it at 30% because of all the bots that keep crawlers out.

How does it find the pages to index?
A. Every website has to register with Google to get indexed
B. Every website has to pay Google to get indexed
C. Google knows which pages to index because of your digital footprint
D. Google has a list of ‘trusted pages’ and it just follows the links from them

How often does a page get visited?
A. Every page that gets indexed is visited every day.
B. It depends on the page, pages like whitehouse.gov get visited daily others rarely
C. Google decides by keeping track of how often pages change.
D. B & C
E. None of the above.

Clickers: What goes into the short index?
A. Every word on the page
B. The words that other sites use when they point to (link to) this page.
C. Only the keywords on the page.
D. Google has a list of special keywords that all pages get indexed by.

Google Short Index & Long Index
- Known as the short barrels and the long barrels...
- Short index:
  - store the words in link texts that point to a page (inbound links!!)
  - the words in a page’s title, and one or two other special things.
- The link text words are attributed to the target page, and not to the page that the link is on.
  - In other words, if my page links to your page, using the link text “Miami hotels”, then the words “Miami” and “hotels” are stored in the short index as though they appeared in your page, but they belong to my page. If 100 pages link to your page, using those same words as link text, then your page will have a lot of entries in the short index for those particular words.
- The long index is used to store all the other words on a page – its actual content.

Understanding your query: Boolean queries
Clickers

* Google responds efficiently to queries by going out and searching the web in real time.

A. TRUE
B. FALSE

How does the index get used at search time by default? (without advanced search)

A. Every website registers with Google exactly which terms to index by and which combinations
B. Google uses “boolean” combinations. The index is made of single words. Google ANDS them together and finds which webpages (URLs) are in the intersection of all the terms
C. Google indexes the pages individually for each person using your digital footprint
D. All of the above.

Make A Query: http://www.google.ca/advanced_search

* When Google gets the query
* It “and”s the two lists together, finding URLs that are on both lists
* It counts them up, records time, shows 10 hits

Processing a query: short and long indices

* First try to get enough results from the short index.
* If you can’t get enough results, them use the long index to add to what they have.

Relevance and Ranking

* Words don’t have to occur together
* Use Boolean queries and quotes
* Logical Operators: AND, OR, NOT
  monet AND water AND lilies “van gogh” OR gauguin vermeer AND girl AND NOT pearl
### How does it rank the pages it finds?

A. It uses what is called a “page rank” algorithm, that uses many different factors  
B. It depends only on who is willing to pay the most.  
C. The government tells it how to rank pages.  
D. None of the above.

### Page Rank Algorithms

- The “crown jewels” of search engines lie in their page rank algorithms.  
- Factors include:  
  - keywords in heading or titles  
  - keyword only in the body text  
  - site is “trustworthy”  
  - links on this page are to relevant pages  
  - links to this page are relevant  
  - age of the page  
  - quality of the text (e.g. absence of misspellings)

### The information that Page Rank uses includes:

- Keywords in heading or titles and keywords in the body text  
- Information about whether the site is “trustworthy”  
- Whether the links on this page are to relevant pages  
- Whether the links to this page are relevant  
- Age of the page  
- Quality of the text (e.g. absence of misspellings)

### Its all free?? : Well no. Who Pays for What?

- Users could pay a subscription fee (early AOL and CompuServe)  
- Web sites could pay for being indexed.  
- The government could pay (taxes?).  
- Advertisers could pay.  
- And it matters who pays cause it affects how it works

### What makes search attractive to advertisers?

A. Advertisements can be targeted more precisely using your digital footprint.  
B. Advertisers only pay when you click on their ad.  
C. Millions of people use search every day.  
D. All of the above.

### Advertiser only pays when you click!

- Advertisers only pay when you click on their ad.
Clickers

* The federal government uses your tax dollars to guarantee that Google and other search engine providers like Microsoft (Bing) return the objectively best results of your query.

A. TRUE
B. FALSE

Clickers

* Web search is free and democratic. Every web page has an equal chance of being indexed and coming out the top of the list.

A. TRUE
B. FALSE

Search engine possible funding models:

A. Users could pay a subscription fee
B. Web sites could pay for being indexed.
C. The government could pay using taxes the same way they pay for roads or police.
D. Advertisers could pay for having their ads featured in the side bar.
E. All of the above are possible.

Search engine current funding model:

A. Users pay a subscription fee
B. Web sites pay for being indexed.
C. The government pays using taxes the same way they pay for roads or police.
D. Advertisers pay for having their ads featured in the side bar.
E. B&D

The Turing Test

* Turing in 1950 published a philosophical paper designed to stop people arguing about whether or not machines could think.
* He proposed that the question be replaced with a test, which is what is now called the Turing Test.
What is the underlying technology called? Question Answering (QA)

This is a Natural Language Processing Technology

I am going to skip over this today, since it is more recent to make sure I have time to cover the stuff we have not reviewed yet. But be sure to look at the clicker questions in here.

Question Answering: A type of NLP

Roots of Question Answering

- Information Retrieval (IR)
- Information Extraction (IE)

Motivated by the observation that Information Retrieval does not answer your question, it provides a ranked list of documents that *might* contain the answer to your question.

Contrast IR/Search with Asking an Expert

Wordnet, Wikipedia, Computing Power, Big Data

<table>
<thead>
<tr>
<th>Natural Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Large volumes of text in many domains</td>
</tr>
<tr>
<td>- Encodes knowledge about meaning and usage of words</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semi-Structured Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Large volumes of data in structured formats</td>
</tr>
<tr>
<td>- Encodes knowledge about the structure of data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NLP (Text Analysis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Entity and Relation Detection</td>
</tr>
<tr>
<td>- Statistical NLP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compute Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Massively parallel computer power</td>
</tr>
<tr>
<td>- High performance computing infrastructure</td>
</tr>
</tbody>
</table>

Enabling Technologies — The Time Was Right

- Natural Knowledge
- Semi-Structured Knowledge
- NLP (Text Analysis)
- Compute Power
What technologies made Watson possible?

A. The Web
B. Compute Power
C. Question Answering
D. Natural Language Processing
E. All of the above.

How long have people been working on it?

- TREC = Text REtrieval Conferences
  - Series of annual evaluations, started in 1992
  - Organized into “tracks”
- Test collections are formed by “pooling”
- Gather results from all participants
- Corpus/topics/judgments can be reused
- TREC has had a QA Track since 1999.
  - http://trec.nist.gov/data/qa/T8_QAdata/development.qa

Types of Question Answering

- Factoid
  - Who discovered oxygen?
  - Where is Ayers Rock?
  - What team won the World Series in 1992?
- List
  - What countries export oil?
  - Name U.S. cities that have a “Shubert” theater.
- Definition
  - Who is Aaron Copland?
  - What is a quasar?

Central Idea of Factoid QA

- Determine the semantic type of the expected answer
  - “Who won the Nobel Peace Prize in 1991?” is looking for a PERSON
- Retrieve documents that have keywords from the question
  - Retrieve documents that have the keywords “won”, “Nobel Peace Prize”, and “1991”
- Look for named-entities of the proper type near keywords
  - Look for a PERSON near the keywords “won”, “Nobel Peace Prize”, and “1991”

Answer Type Hierarchy
**Factoid QA**

* The central idea of Factoid questions is to determine the semantic type of the expected answer. Some of the possible semantic types of expected answers are:

A. Persons and Places
B. Distances, Heights, Lengths and other numerical values
C. Universities, Businesses, Organizations
D. Types of Animals, such as Reptile or Mammal
E. All of the above

**The types of questions in question answering research are:**

A. Definition Questions: e.g. Who is Jared Borgetti?
B. Factoid Questions: e.g. How many feet above sea level is the UCSC Campus?
C. List Questions: e.g. Name the players on Brazil's World Cup Winning Team in 2002.
D. None of the above
E. All of the above

**An Example**

**Who won the Nobel Peace Prize in 1991?**

But many foreign investors remain sceptical, and western governments are withholding aid because of the Slorc’s dismal human rights record and the continued detention of Ms Aung San Suu Kyi, the opposition leader who won the Nobel Peace Prize in 1991.

The military junta took power in 1988 as pro-democracy demonstrations were sweeping the country. It held elections in 1990, but has ignored their result. It has kept the 1991 Nobel peace prize winner Aung San Suu Kyi – leader of the opposition party which won a landslide victory in the poll – under house arrest since July 1989.

The regime, which is also engaged in a battle with insurgents near its eastern border with Thailand, ignored a 1990 election victory by an opposition party and is detaining its leader Ms Aung San Suu Kyi, who was awarded the 1991 Nobel Peace Prize. According to the British Red Cross, 5,000 or more refugees, mainly the elderly and women and children, are crossing into Bangladesh each day.

Who won the Nobel Peace Prize in 1991?

**First finding: Cannot Anticipate Answer Types!!**

This graph from Prof. Walker’s lecture on IBM’s Watson was used to demonstrate that:

A. It is always possible to determine the semantic type of the expected answer
B. Anticipating the question types and building specialized databases would not let Watson win at Jeopardy
C. Most questions have to do with Distances, Heights, Lengths and other numerical values
D. Online encyclopedias have most answers already pre-computed in an easy to extract format
E. None of the above

**Interesting tradeoff:** Knowledge, Precision, Open Domain
Wikipedia: What knowledge can we get from Wikipedia?


Using WordNet: Online Thesaurus.

- http://wordnetweb.princeton.edu/
- What is the service ceiling of a U-2?
- Can access it FROM a program (not just this interface).

What kind of information does WordNet have?

Using WordNet: Online Thesaurus.

- What is the service ceiling of a U-2
- Can access it as a program.

Does WordNet know about Chocolate Cake?

Chocolate Decadence Cake?: The Web knows all!
Structured knowledge vs. text search (clicker)

- What does this graph show?
  A. Structured knowledge sources can answer most of the questions with high confidence
  B. Text search by itself can never answer more than 30% of the questions
  C. Structured knowledge by itself can answer at least 50% of the questions with 50% confidence
  D. Watson must combine deep (natural language understanding) and shallow (words and indices) semantic analysis to win

A Few Guiding Principles

- Specific large hand-crafted models won’t cut it
  – Too slow, too narrow, too brittle, too biased
    – Need to acquire and analyze information from on-line knowledge sources
- Intelligence from many diverse methods
  – Many diverse algorithms must be combined: No single one is expected to solve the whole problem. Each addressing different weaknesses.
  – Relative impact of many overlapping methods must be learned
- Massive Parallelism is a Key Enabler
  – Pursue many competing independent hypotheses over large data
  – Efficiency will demand simultaneous threads of evidence evaluation

This is not easy!

Evaluating possibilities and their Evidence

Potential Business Applications

Creativity in the kitchen?

http://www.psfk.com/2013/03/ibm-watson-top-chef.html
The potential business applications of QA are:

A. Customer care and help systems
B. Answering Healthcare questions
C. A & B
D. Playing Jeopardy, which can’t make IBM much money
E. None of the above

People try to design algorithms e.g. to sort things, to run quickly

Algorithms

- **Def.** *An algorithm is a precise, systematic process for an agent to produce a specified result*
- Programs are algorithms
- Five properties characterize algorithms
  - **Input specified** – tell form and amount of input required
  - **Output specified** – tell form and amount of output produced
  - **Definiteness** – say explicitly what to do & in what order
  - **Effectiveness** – operations within agent’s abilities
  - **Finiteness** – will stop and give an answer or say “none”

Searching

- Guess a number between 1 and 100. How many guesses do you need?
- Scatter a deck of cards on the floor. How many do you have to turn over to find the ace of spades?
Linear = Time Proportional To N

- Problems whose work (computation time) is proportional to n are called n-time or linear time problems
  - Making an image lighter in your photo software
  - Adding a column of numbers in a spreadsheet
  - Crawling the Internet looking for links
  - ... many more ... linear problems are common
- Some problems are not ...

Time proportional to $n^2$ = Polynomial time

- Other computations have running time
  - proportional to $n^2$ – matrix multiplication
  - proportional to $n^4$
- ... All of them are lumped together as "polynomial time computations"
  - Considered to be realistic ... a person can wait
  - Polynomial, but not linear
  - Many algorithms you would learn in 12B, 101

Finding an item in an unsorted list

What is the time complexity of finding an item in an unsorted list?

A. Less than linear (log(n) or constant)
B. Linear
C. Polynomial but more time than linear
D. NP (exponential like $2^n$)

Sorting

- Putting a sequence of items into alphabetical or numerical order
- First: let's try exchanging them as we go along (bubbling them along)
  - walrus seal whale gull clam
- Algorithm: compare to all following items, reorder if needed
- Other ways to sort we will talk about in a minute

How Long To Sort w/ Bubble Sort?

- The pattern is, for $n$ items
  n-1 focus on first item in the list
  n-2 focus on second item
  n-3 focus on third item
  ...
  1 on next to last
- n-2 rows in list, we just want a good estimate
  - average of each row $\frac{n}{2}$
  - so (n-2) times each row
  - Multiply by average = $\frac{n^2}{2}$
  - Computing time = Time proportional to $n^2$

There are Different Algorithms

- Is there a better way to do sorting?

  QUICKSORT
  - Fastest known sorting algorithm in practice
  - Average case: O(N log N) (we don’t prove it)
  - Worst case: O(N^2)
  - But, the worst case seldom happens.
  - A divide-and-conquer recursive algorithm
  - Video of selection vs quicksort: http://youtu.be/cVMKXKoGu_Y
Quicksort is the best: Divide and Conquer

- Divide step:
  - Pick any element (pivot) \( v \) in \( S \)
  - Partition \( S - \{v\} \) into two disjoint groups
    
    \[ S_1 = \{x \in S - \{v\} | x \leq v\} \]
    
    \[ S_2 = \{x \in S - \{v\} | x > v\} \]

- Conquer step: recursively sort \( S_1 \) and \( S_2 \)

- Combine step: the sorted \( S_1 \) (by the time returned from recursion), followed by \( v \), followed by the sorted \( S_2 \) (i.e., nothing extra needs to be done)

To simplify we may assume that we don’t have repetitive elements, so to ignore the “equality” case!

Where does the \( \log N \) come from?

Binary Search

- Pick a number between 1 and 1000. How many guesses will I need?

  - Depends if it’s sorted.
  - Web Search: What about that index with 25 million entries?

Finding an item in a sorted list

What is the time complexity of finding an item in a sorted list?

A. Less than linear (\( \log(n) \) or constant)
B. Linear
C. Polynomial but more time than linear
D. \( \text{NP} \) (exponential like \( 2^n \))
E. Unsolvable (like the halting problem)

Which grows slowest?

A. Time proportional to \( N^2 \)
B. Time proportional to \( N \log(N) \)

To Infinity And Beyond

- There are more complex computations ...
  - Suppose you want to visit all cities in the US (for a rock concert?) and you want to minimize your how much you pay for airplane tickets
  - You could select an ordering of cities (SEA \( \rightarrow \) PDX \( \rightarrow \) SFO \( \rightarrow \) LAX ...), and compute the ticket price.
  - Then pick another ordering (SEA \( \rightarrow \) SFO \( \rightarrow \) LAX \( \rightarrow \) PDX ...), compute this ticket price and compare to the previous one
  - Always keep the cheapest itinerary

  - This seems very dumb ... is there a better way?

Traveling Salesman Problem

- Actually, no one knows a way to solve this problem significantly faster than checking all routes and picking the cheapest...
- Not polynomial time ... so we are guessing that there is no polynomial solution (Non Polynomial = \( \text{NP} \))
- This is what is called an \( \text{NP-Complete problem} \)
  - Many many related problems ... the best solution is “generate and check”
    - Best way to pack a container ship
    - Most efficient scheduling for high school students’ classes
    - Least fuel to deliver UPS packages in Washington
    - Fewest public alert broadcast stations for US
Which grows fastest?

A. Time proportional to \(N^2\)
B. Time proportional to \(N \log(N)\)
C. Time proportional to \(N\)
D. Time proportional to \(2^N\)

Summary: Computational Complexity

- Many computations have time proportional to \(n\)
- Many, like sort, have running time proportional to \(n^2\)
- Others have running time proportional to \(n^3, n^4, \ldots\)
- Some computations are computable in principle but not in practice: NP-complete
- Some things cannot be computed at all, such as the Halting Problem

White, Gray, Black

- You know that gray is just different degrees of white as the “light is turned down” till we get to black
- Black = \([0, 0, 0]\)
- Gray = \([128, 128, 128]\)
- White = \([255, 255, 255]\)
- White, gray, black all have same values for RGB

Colors

Colors use different combinations of RGB
- Purple
  - Red=160
  - Green=76
  - Blue=230

Random Numbers

- Random numbers should be called random number sequences, because the definition requires that no matter how many numbers you already know in the sequence, it’s not possible to predict the next one. A non-random sequence is 2, 4, 6, 8, 10, …
- Computers cannot produce random numbers (because computers are completely predictable), but they can produce a sequence of numbers that passes all of the tests for randomness. These are called pseudo-random numbers, but everyone drops the “pseudo” part.
- To generate a random number in Processing we write:
  - \(\text{random(smallest possible number, largest possible number)}\)
  - \(\text{random(smallest possible number, largest possible number)}\)
  - \(\text{random(0, 255)}\)
  - \(\text{random(0, 1)}\)
What would we do in Processing? What does Random do?

```java
void draw() {
    // code...
}
```

Let's Try it!

```java
void draw() {
    // code...
}
```

Positional Notation

- The RGB intensities are binary numbers
- Binary numbers, like decimal numbers, use place notation

\[
1101 = 1 \times 1000 + 1 \times 100 + 0 \times 10 + 1 \times 1
\]

except that the base is 2 not 10

\[
1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1
\]

\[
= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0
\]

1101 in binary is 13 in decimal

The Red of P As A Binary Number

```
Purple: Red=160, Green=76, Blue=230
```

```
Given a binary number, add up the powers of 2 corresponding to 1s
```

```
\[
1 \times 2^7 = 1 \times 128 = 128
0 \times 2^6 = 0 \times 64 = 0
1 \times 2^5 = 1 \times 32 = 32
0 \times 2^4 = 0 \times 16 = 0
0 \times 2^3 = 0 \times 8 = 0
0 \times 2^2 = 0 \times 4 = 0
0 \times 2^1 = 0 \times 2 = 0
0 \times 2^0 = 0 \times 1 = 0
\]

\[
= 160
\]

Green of P As A Binary Number

```
Purple: Red=160, Green=76, Blue=230
```

```
Given a binary number, add up the powers of 2 corresponding to 1s
```

```
\[
0 \times 2^7 = 0 \times 128 = 0
1 \times 2^6 = 0 \times 64 = 0
0 \times 2^5 = 1 \times 32 = 32
0 \times 2^4 = 0 \times 16 = 0
1 \times 2^3 = 0 \times 8 = 0
1 \times 2^2 = 0 \times 4 = 0
0 \times 2^1 = 0 \times 2 = 0
0 \times 2^0 = 1 \times 1 = 1
\]

\[
= 76
\]

What if we want it to be GREENER! BIGGER NUMBER OR SMALLER?

1. BIGGER BINARY NUMBER for GREEN of RGB
2. SMALLER BINARY NUMBER for GREEN of RGB

Clicker

- What if we want it to be GREENER! BIGGER NUMBER OR SMALLER?
What about making a redder Purple?

- So Purple is (160, 76, 230) which is
  - 1010 0000
  - 0110 1100
  - 1110 0110
  - (160, 76, 230)

Suppose you decide it’s not “red” enough
- Increase the red by 16 = 1 0000
  - 1010 0000
  - + 1 0000
  - 1011 0000

You already know how to add in binary, right?

How do we make a Redder Purple

- ADD 16 more
  - 0011 0000
  - 1011 0000
  - + 1 0000
  - 1100 0000

You already know how to add in binary, right?