Course Description

“IBM Watson is a computer system capable of answering rich natural language questions and estimating its confidence in those answers at a level of the best humans at the task. On Feb. 14-16, 2011, in an televised event, Watson triumphed over the best human players of all time on the American quiz show, Jeopardy!. In this course we will discuss the main principles of natural language processing, computer representation of knowledge and discuss how Watson solved some of its answers (right and wrong).”
Course Focus

- Unstructured Information Managing Architecture UIMA (in Java)
- Natural Language Processing (NLP)
- Knowledge Representation (KR) in Prolog
Instructor Information

- Dr. Paul Fodor
  1437 Computer Science Building

- Office hours: Tuesdays 10:00AM-11:30AM &
  Wednesdays 8:00AM-9:30AM

- Email: pfodor (at) cs (dot) stonybrook (dot) edu

- Please include “ITS 102” in the email subject and your name in your email correspondence
General Information

• Meeting Information:
  • Lectures: Wednesdays, 11:30AM - 12:50PM, 11-week option, Computer Science Building 2116.
  • Course Web page:
  • Blackboard will also be used for assignments, grades and course material.
Textbook

- **No textbook is required.**
- We will use material from:
Coursework

- **Grading Schema:**
  - Students will be evaluated on the basis of homework and lab work, participation in discussion of lecture materials, and interaction with faculty and other students.
  - Because of the variety of offerings, it is not possible to specify precise breakdowns of the value of each type for all sections.

  - *Class Participation:* Students are expected to contribute their own ideas and to ask questions during class.
  - *Class Attendance:* Students are expected to attend all of the class sessions for this seminar.
  - *Assignments:* there will be short homeworks and class assignments.
  - *Required Reading:* before each class there will be required reading relevant to that class.
Academic Integrity

- You can discuss general assignment concepts with other students.
- You MAY NOT share assignments, source code or other answers:
  - Assignments are subject to manual and automated similarity checking.
- If you cheat, you MAY be brought up on academic dishonesty charges without warning - we follow the university policy:
  - [http://www.stonybrook.edu/uaa/academicjudiciary](http://www.stonybrook.edu/uaa/academicjudiciary)
Please

- Please be on time,
- Please show respect for your classmates,
- Please turn off (or use vibrate for) your cellphones.
- On-topic questions are welcome.
Real Language is Real Hard

- **Chess**
  - A finite, mathematically well-defined search space
  - Limited number of moves and states
  - Grounded in explicit, unambiguous mathematical rules

- **Human Language**
  - Ambiguous, contextual and implicit
  - Grounded only in human cognition
  - Seemingly infinite number of ways to express the same meaning
The Best Human Performance: Our Analysis Reveals the Winner’s Cloud

Each dot represents an actual historical human Jeopardy! game

Top human players are remarkably good.

Computers?

Winning Human Performance

2007 QA Computer System

More Confident

Less Confident

Grand Champion Human Performance

(c) Paul Fodor (CS Stony Brook)
DeepQA: The Technology Behind Watson
Massively Parallel Probabilistic Evidence-Based Architecture

Generates and scores many hypotheses using a combination of 1000’s Natural Language Processing, Information Retrieval, Machine Learning and Reasoning Algorithms. These gather, evaluate, weigh and balance different types of evidence to deliver the answer with the best support it can find.
How Watson Processes a Question

Keywords: 1698, comet, paramour, pink, ...
AnswerType(comet discoverer)
Date(1698)
Took(discoverer, ship)
Called(ship, Paramour Pink)
...

Candidate Answer Generation

1) Edmond Halley (0.85)
2) Christiaan Huygens (0.20)
3) Peter Sellers (0.05)

Merging & Ranking

Evidence Scoring

(c) Paul Fodor (CS Stony Brook)
Apache UIMA

- Open-source framework and tools for building NLP applications
- Key Concepts
  - *Common Analysis Structure (CAS)*: Container for Data Structures in user-defined data model (which can be defined in UML)
  - *Annotator*: Pluggable component (Java or C++, among others) that reads and writes a CAS
  - *Aggregate Analysis Engine*: Collection of Annotators
Watson in UIMA

Aggregate Analysis Engine: Question/Topic Analysis
- Tokenizer
- NED
- Deep Parser
- Predicate Argument Structure
- Shallow SRD
- Focus and LAT Detection
- Anaphora Resolution
- Decomposition and Classification
- Deep SRD

Watson Top-Level Aggregate Analysis Engine
- Question/Topic Analysis
- Question Decomposition
- Hypothesis Generation
- Hypothesis and Evidence Scoring
- Synthesis
- Final Merging & Ranking

(c) Paul Fodor (CS Stony Brook)
Natural Language Processing In Watson

Text (Question or Evidence) →

- Tokenization
- Deep Parsing
- Predicate Argument Structure
- Named Entity Recognition

Predicate Argument Structure Graph

- Rule-Based and Statistical Pattern Matching
- Relations
- Co-Reference Resolution
- Question Focus
- Lexical Answer Types (LATs)
- Question Classification

(c) Paul Fodor (CS Stony Brook)
POETS & POETRY: He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.
PAS and Relations in a Supporting Passage

"Songs of a Sourdough" is a poetry book by Robert W. Service.

be_p_23

Songs_of_a_Sourdough_p_1

Songs_of_a_Sourdough_p_1

"SameAs_target"_altNames_name

song_p_1

book_p_35

poetry_p_28

by_p_40

Robert_W_Service_p_43

Robert_W_p_43

mod_qtsl

mod_nbj

mod_unm

objprep

of_p_7

of_p_7

sourdough_p_12

nnModification_modifer

objprep

dm_arg_arg

dm_obj_arg

Robert_W_Service_p_43

nnModification_modifer

dm_arg_arg

dm_subj_arg

Robert_W_p_43

mod_chsl

mod_adj

service_p_53
Unstructured Information Management Architecture (UIMA)

- Platform independent standard for interoperable text and multi-modal analytics

**UIMA Annotation**

Document text:
“...seminar in GN-K35 on October 24, 2007”
Our work in IBM Watson - UIMCA CAS Prolog Interface Architecture

QParse 2 Analysis Engine

CAS Facts

Focus

Modifiers

Answer-type

Rules

Prolog

Focus,
Answer-type,
Modifier
Annotation
Types

UIMA CAS Focus&Answer-type Annotator

casToProlog(Cas)

retrievedPrologAnnotations(Cas)

New Annotations

XSG Parser,
Entity,
Relation,
Predicate-
Argument-
Structure
Annotators

UIMA Pipeline

(c) Paul Fodor (CS Stony Brook)
Focus Computation rules

- The focus is the “node“ that refers to the unspecified answer
  - “What is the name of the airport in Dallas?”
    - Focus = “airport“
  - “What is the population of Iceland?”
    - Focus = “population”
- The focus abstracts different syntactical constructs:
  1) What X …
  2) What is the X that…
  3) Which of the X …
  4) What is the name of the X that…
  5) Name the X that…
  …
- Applications:
  - Answer-type detection
  - Logical form answer-selection
“How much/many” rule:

- Pattern: HOW_MANY/MUCH X VERB …?
- Examples:
  
  “How many hexagons are on a soccer ball?”
  “How much does the capitol dome weigh?”
  “How much folic acid should an expectant mother get daily?”

```
focus(QuestionRoot, [Determiner]):-
  getDescendantNodes(QuestionRoot,Determiner),
  lemmaForm(Determiner,DeterminerString),
  howMuchMany(DeterminerString),!.
  % "how much/many", "this much",…
  ```
Example QParse2 Focus Detection Rules

- “What is X …” rule:
  - Pattern: WHAT IS X …?
  - Example:
    - “What is the democratic party symbol?”
    - “What is the longest river in the world?”

```prolog
focus(QuestionRoot, [Pred]):-
  getDescendantNodes(QuestionRoot, Verb),
  lemmaForm(Verb, "be"),
  subj(Verb, Subj),
  lemmaForm(Subj, SubjString),
  whatWord(SubjString), % e.g., "what", "which" ("this", "these")
  pred(Verb, Pred),!.
```
Answer-type Computation Rules

- Heuristics to compute the type of the answer

  Focus lexicalization (lexical chains using Prolog WordNet followed by a mapping to our taxonomy)

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>What American revolutionary general turned over West Point to the British?</td>
<td>[com.ibm.hutt.MilitaryLeader]</td>
</tr>
</tbody>
</table>

  **Table lookup for the verb:**

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
</table>

  **Table lookup for the focus:**

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>How far is it from the pitcher's mound to home plate?</td>
<td>[com.ibm.hutt.Length]</td>
</tr>
<tr>
<td>When was Lyndon B Johnson president?</td>
<td>[com.ibm.hutt.Year]</td>
</tr>
</tbody>
</table>

  **Table lookup for the focus (noun) + the verb:**

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>What instrument measures radioactivity?</td>
<td>[com.ibm.hutt.Tool]</td>
</tr>
</tbody>
</table>
### Answer-type Computation Rules

- Cascading rules in order of generality
  - first rule that fires returns the most specific answer-type for the question

**Look at the focus + verb:**

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>How much did Marilyn Monroe weigh?</td>
<td>[com.ibm.hutt.Weight]</td>
</tr>
<tr>
<td>How much did the first Barbie cost?</td>
<td>[com.ibm.hutt.Money]</td>
</tr>
</tbody>
</table>

**Look at the focus + noun:**

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many Earth days does it take for Mars to orbit the sun?</td>
<td>[com.ibm.hutt.Duration]</td>
</tr>
</tbody>
</table>

**Look only at the focus:**

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many moons does Venus have?</td>
<td>[com.ibm.hutt.WholeNumber]</td>
</tr>
<tr>
<td>How much calcium is in broccoli?</td>
<td>[com.ibm.hutt.Number]</td>
</tr>
</tbody>
</table>
Example QParse 2 Answer-type Detection Rules

- Time rule (e.g. when):
  Pattern: WHEN VERB OBJ; OBJ VERB THEN
  Example: **When** was the US capitol **built**?
 .answerType => ["com.ibm.hutt.Year"]

```prolog
answerType(_QuestionRoot,FocusList,timeAnswerType,ATList):-
  member(Mod,FocusList),
  lemmaForm(Mod,ModString),
  wh_time(ModString), % "when", "then"
  whadv(Verb,Mod),
  lemmaForm(Verb,VerbString),
  timeTableLookup(VerbString,ATList),!.
```
Example QParse 2 Answer-type Detection Rules

- "How … VERB" rule:
  Pattern: How … VERB?
  Example: “How did Virginia Woolf die?”
  ```prolog
  answerType => ["com.ibm.hutt.Disease",
                  "com.ibm.hutt.MannerOfKilling",
                  "com.ibm.hutt.TypeOfInjury"]
  answerType(_QuestionRoot,FocusList,howVerb1,ATList):-
    member(Mod,FocusList),
    lemmaForm(Mod,"how"),
    whadv(Verb,Mod),
    lemmaForm(Verb,VerbString),
    howVerbTableLookup(VerbString,ATList), !.
  ```
QParse2 Evaluation

- 370 correct matches with the standard (89.5%)

343 exact answer-type (83%):

<table>
<thead>
<tr>
<th>Question</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who created the literary character Phineas Fogg?</td>
<td>[com.ibm.hutt.ContentCreator]</td>
</tr>
<tr>
<td>What is the name of the airport in Dallas Ft Worth?</td>
<td>[com.ibm.hutt.Facility]</td>
</tr>
<tr>
<td>What city is Disneyland in?</td>
<td>[com.ibm.hutt.City]</td>
</tr>
<tr>
<td>What color belt is first in karate?</td>
<td>[com.ibm.hutt.Color]</td>
</tr>
</tbody>
</table>

27 of the correct matches were NounPhrase (6.5%): one cannot determine the type (unless he already knows the answer of the question)

<table>
<thead>
<tr>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did Peter Minuit buy for the equivalent of 2400?</td>
</tr>
<tr>
<td>What is the gift for the 20th anniversary?</td>
</tr>
<tr>
<td>What did Ozzy Osbourne bite the head off of?</td>
</tr>
</tbody>
</table>

No type in our taxonomy

<table>
<thead>
<tr>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the word which means one hiring his relatives?</td>
</tr>
<tr>
<td>What is a word spelled the same backward and forward called?</td>
</tr>
</tbody>
</table>
## QParse2 Evaluation

- 3 results had a subset of the manually annotated answer types

<table>
<thead>
<tr>
<th>Question</th>
<th>Standard Answer Type</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
</table>

- 17 results had extra types than (a superset of) the manually annotated answer types

<table>
<thead>
<tr>
<th>Question</th>
<th>Standard Answer Type</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long before bankruptcy is removed from a credit report?</td>
<td>[com.ibm.hutt.Duration]</td>
<td>[com.ibm.hutt.Duration, com.ibm.hutt.Length]</td>
</tr>
<tr>
<td>How long is a quarter in an NBA game?</td>
<td>[com.ibm.hutt.Duration]</td>
<td>[com.ibm.hutt.Duration, com.ibm.hutt.Length]</td>
</tr>
</tbody>
</table>

- 6 results had a super-type of the manually annotated answer types

<table>
<thead>
<tr>
<th>Question</th>
<th>Standard Answer Type</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>When did International Volunteers Day begin?</td>
<td>[com.ibm.hutt.Year]</td>
<td>[com.ibm.hutt.DateTime]</td>
</tr>
</tbody>
</table>
QParse2 Evaluation

- 23 results different than the standard manual annotation:
  - Need for more answer-type detection rules

<table>
<thead>
<tr>
<th>Question</th>
<th>Standard Answer Type</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
</table>

- WordNet word sense disambiguation algorithm

- Wrong Parse

<table>
<thead>
<tr>
<th>Question</th>
<th>Standard Answer Type</th>
<th>QParse 2 AnswerType</th>
</tr>
</thead>
<tbody>
<tr>
<td>What 20th century American president died at Warm Springs, Georgia?</td>
<td>[com.ibm.hutt.President]</td>
<td>[com.ibm.hutt.Date]</td>
</tr>
</tbody>
</table>
Results!