Watson, Design & NLP

CSC490: Cognitive Computing

Next Milestone: Design

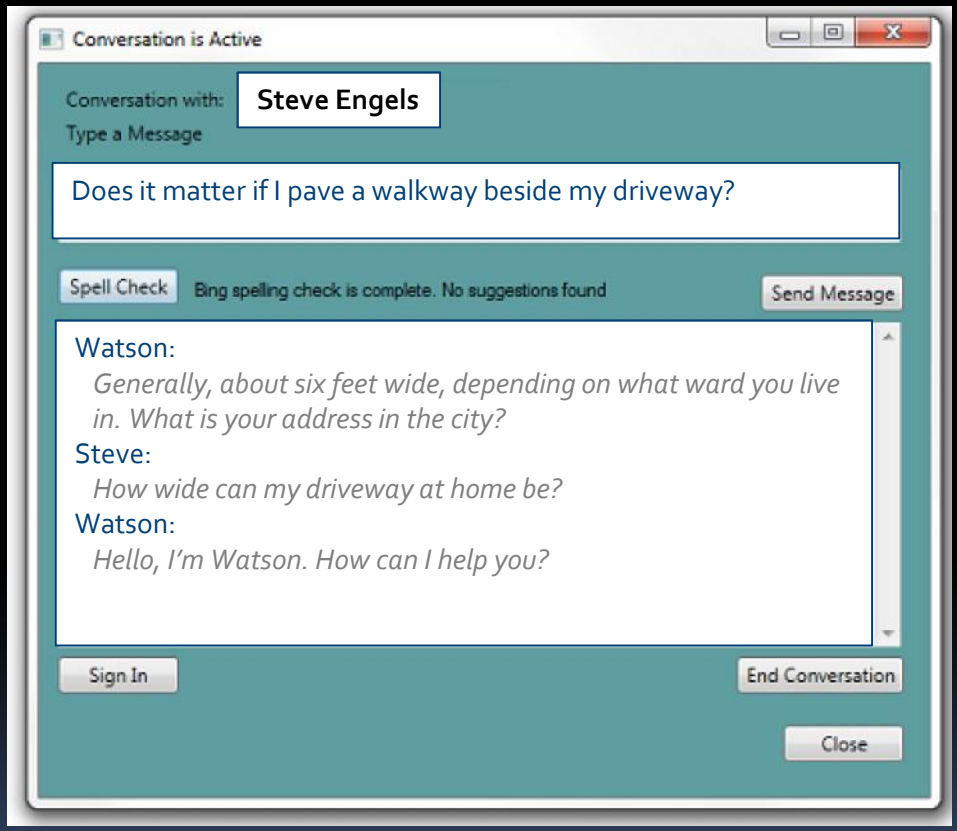
- Basically, everything short of code.
  - use cases
  - software structure diagrams
  - product backlog
  - scheduling diagrams
  - interface designs
- Make sure you outline examples of how Watson will be used to create your product's core functionality.

Example: City Planner

- Basic Pitch:
  - Using Watson to help new residents and businesses navigate city bylaws.

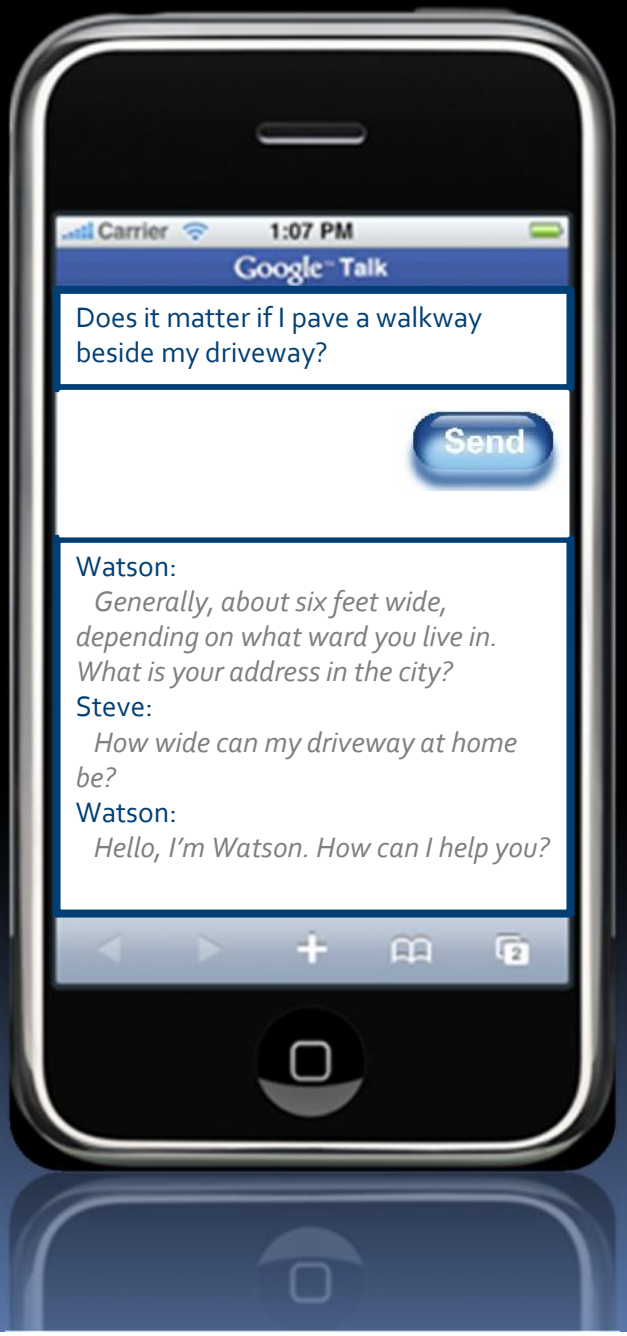
Example: City Planner

- Design:
  - Web interface: a single text entry box for questions and a scrollable text window for the past conversation with Watson.



City Planner

- Design:
  - Mobile interface: a single text entry box for questions and a scrollable text window for the past conversation with Watson.
  - Server side: provide interface between user questions and Watson API



Server Implementation

- Performs following steps:
  - Receive question from website / app
  - Compose question for Watson
  - Send composed question to Watson
  - Receive response passage
  - Parse response
  - Display parsed response

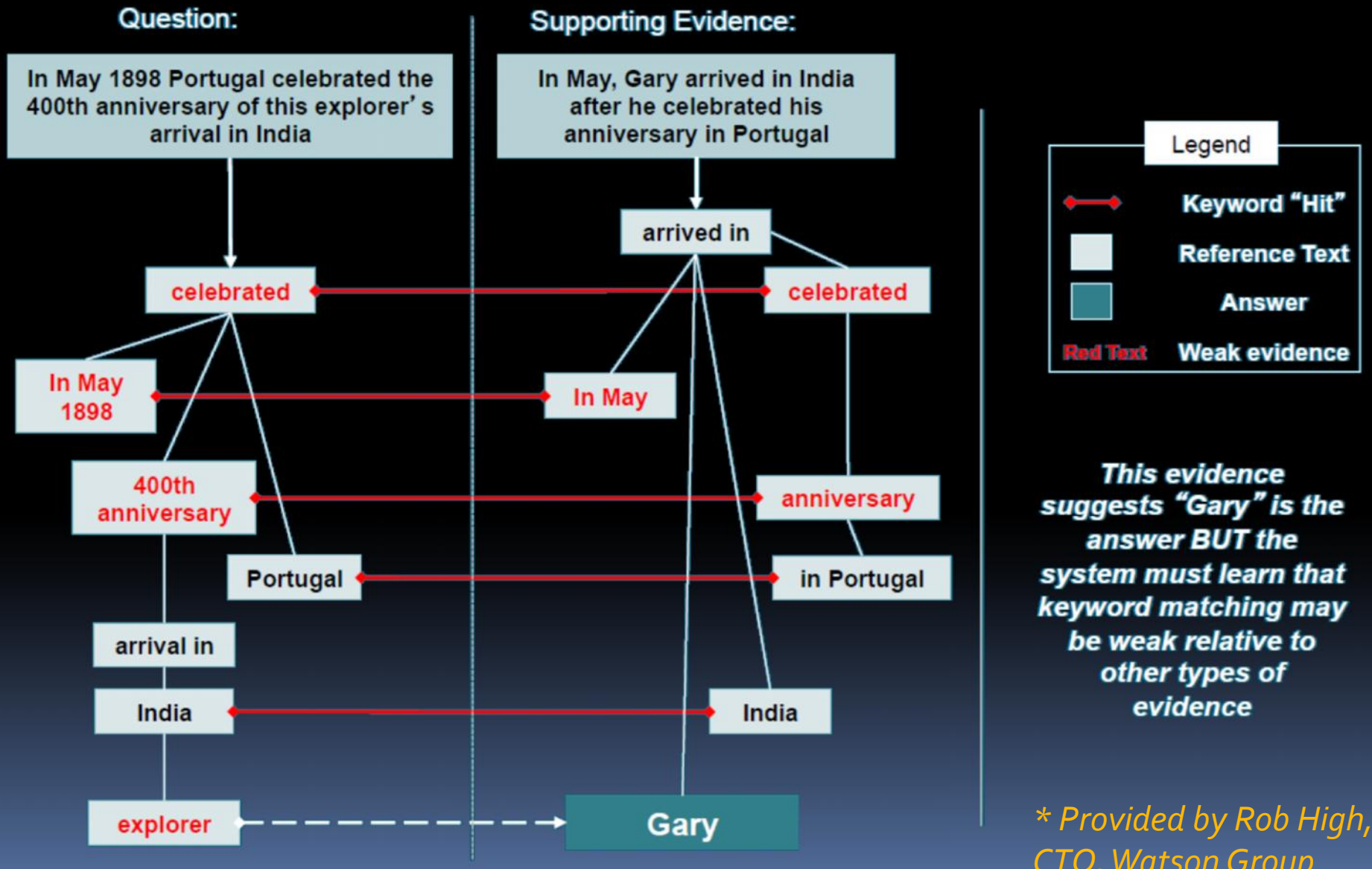
Natural Language Processing

What you'll need

- All of the NLP will fall into these pre- and post-processing stages.
- Handling these stages is mostly a simple data processing task, with some NLP techniques and tools tossed in as well.
- First, let's understand Watson….

What Watson is doing

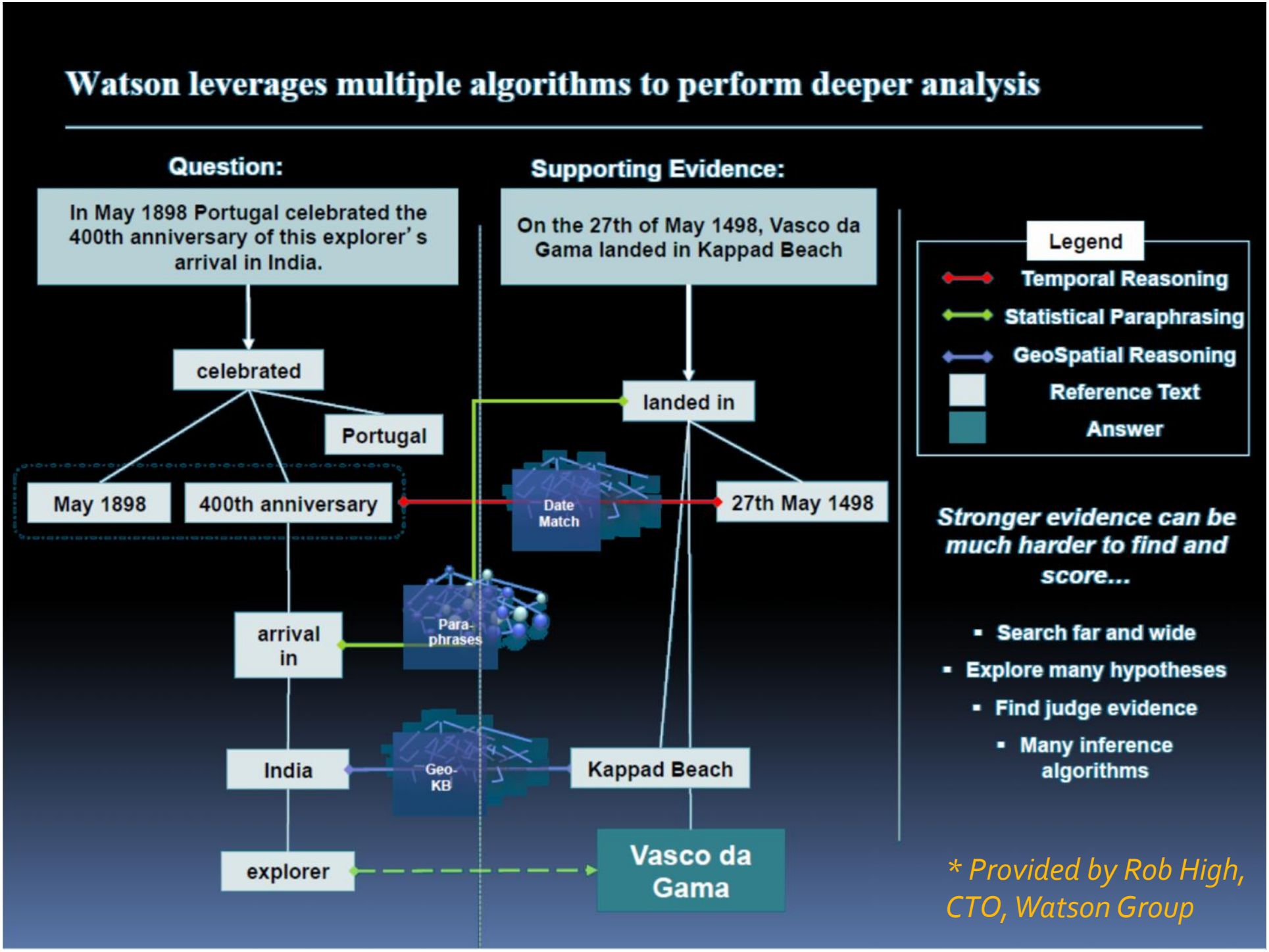
Answering complex questions requires more than keyword evidence



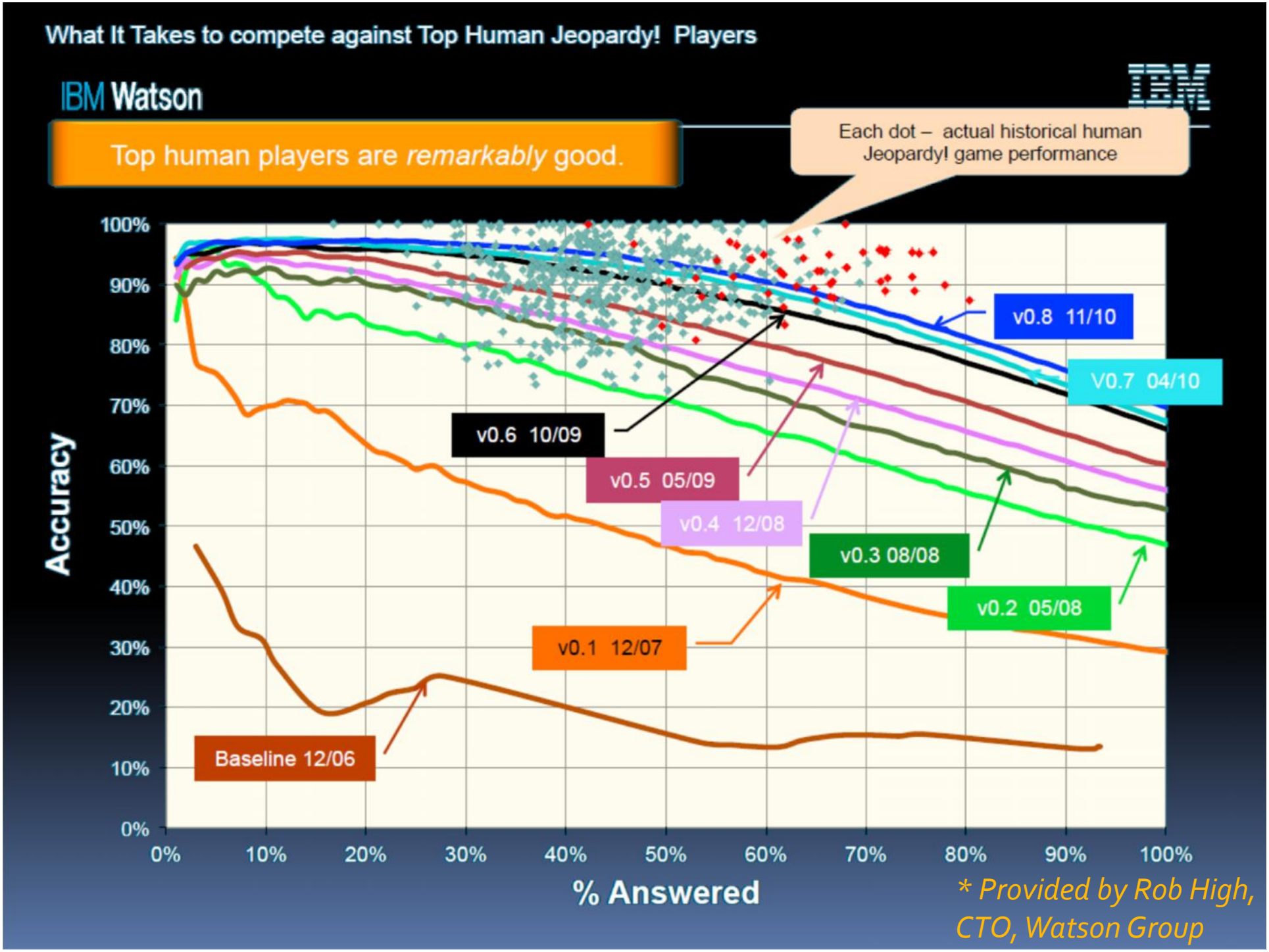
This evidence suggests "Gary" is the answer BUT the system must learn that keyword matching may be weak relative to other types of evidence

** Provided by Rob High, CTO, Watson Group*

Watson leverages multiple algorithms to perform deeper analysis

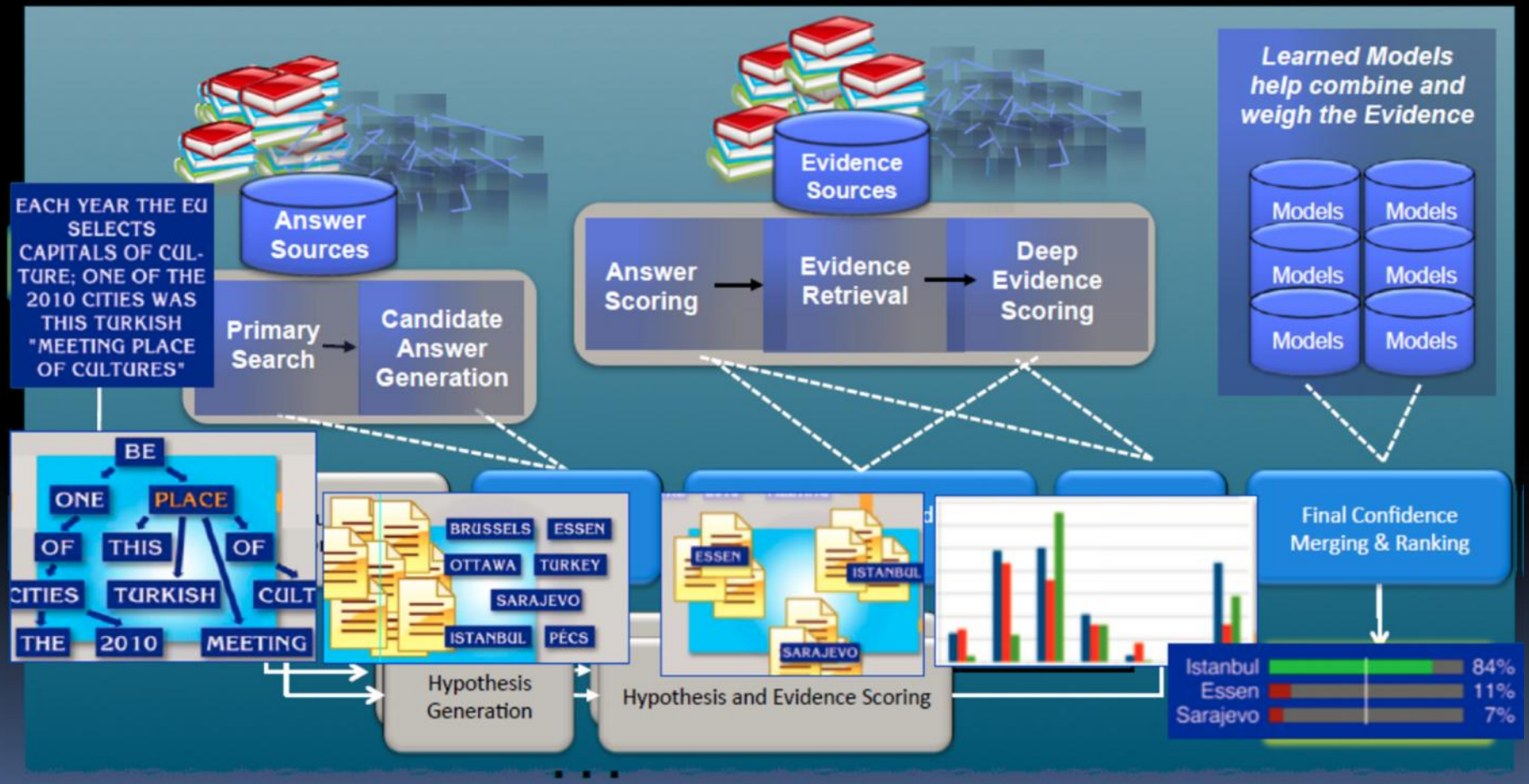


** Provided by Rob High, CTO, Watson Group*


What It Takes to compete against Top Human Jeopardy! Players
IBM Watson
IBM
Top human players are *remarkably* good.
Each dot – actual historical human Jeopardy! game performance
v0.8 11/10
V0.7 04/10
v0.6 10/09
v0.5 05/09
v0.4 12/08
v0.3 08/08
v0.2 05/08
v0.1 12/07
Baseline 12/06
Accuracy
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
% Answered
* Provided by Rob High, CTO, Watson Group

DeepQA: The architecture underlying Watson

Generates many hypotheses, collects a wide range of evidence and balances the combined confidences of over 100 different analytics that analyze the evidence from different dimensions



** Provided by Rob High, CTO, Watson Group*

Using the tool

NLP Basics

- Whenever possible, use pre-existing tools.
  - Don't try to solve natural language processing!
- Consider customizing your application to take advantage of what Watson gives you.

The General NLP Problem

- User questions are unreliable sources of input:
  - Often ungrammatical (particularly speech)
  - Anaphora / coreference
  - Implied action
- Watson's responses will be in passage form:
  - Jeopardy-style answers are not supported
  - Extraction of answers from passages may require syntactic or semantic understanding of text.
- Note: The major tasks have been mostly solved.
  - Accuracy rate: ~ 95% - 98%

NLP Tools

- The Stanford Parser:
  - Breaks sentences down into component parts-of-speech (e.g. nouns, noun phrases, etc)

"The strongest rain ever recorded in India shut down the financial hub of Mumbai, snapped communication lines, closed airports and forced thousands of people to sleep in their offices or walk home during the night, officials said today."

The/DT strongest/JJS rain/NN ever/RB recorded/VBN in/IN India/NNP shut/VBD down/RP the/DT financial/JJ hub/NN of/IN Mumbai/NNP ,/, snapped/VBD communication/NN lines/NNS ,/, closed/VBD airports/NNS and/CC forced/VBD thousands/NNS of/IN people/NNS to/TO sleep/VB in/IN their/PRP$ offices/NNS or/CC walk/VB home/NN during/IN the/DT night/NN ,/, officials/NNS said/VBD today/NN ./.

NLP Tools

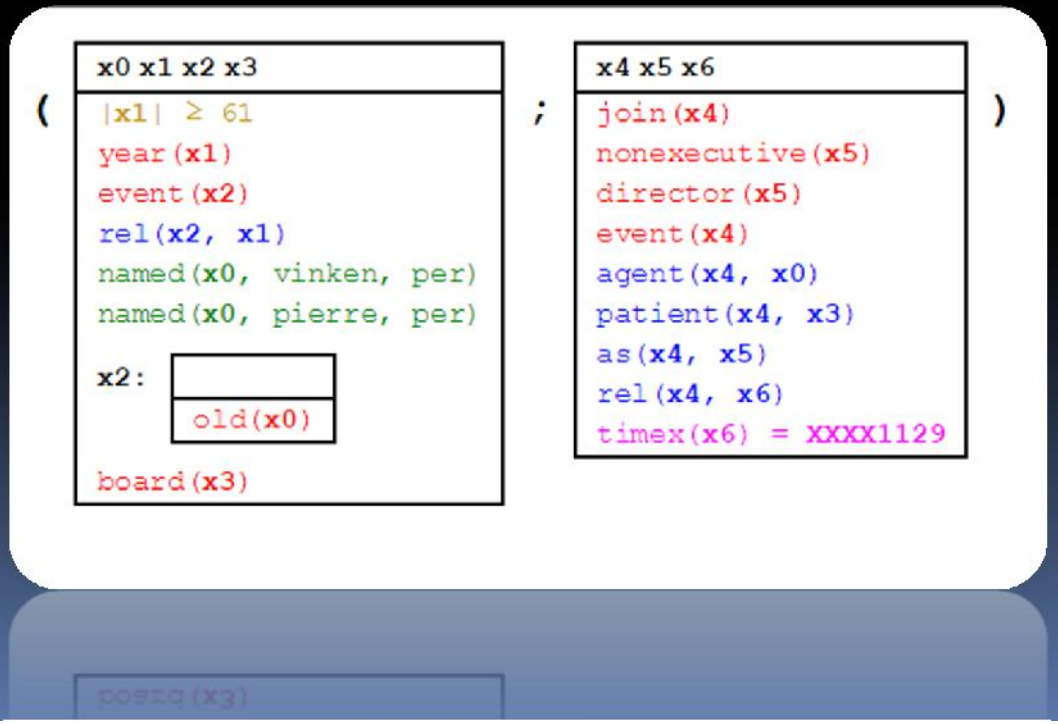
- The Stanford Parser:
  - Parse tree output:
    - Shows how sentence is broken down into components.
    - Useful for showing how certain words are connected to others.
  - Demo:
    - http://nlp.stanford.edu:8080/parser/index.jsp

```
(ROOT
  (S
    (S
      (NP
        (NP (DT The) (JJS strongest) (NN rain))
        (VP
          (ADVP (RB ever))
          (VBN recorded)
          (PP (IN in)
            (NP (NNP India)))))
      (VP
        (VP (VBD shut)
          (PRT (RP down))
          (NP
            (NP (DT the) (JJ financial) (NN hub))
            (PP (IN of)
              (NP (NNP Mumbai)))))
        (, ,)
        (VP (VBD snapped)
          (NP (NN communication) (NNS lines)))
        (, ,)
        (VP (VBD closed)
          (NP (NNS airports)))
        (CC and)
        (VP (VBD forced)
          (NP
            (NP (NNS thousands))
            (PP (IN of)
              (NP (NNS people))))
          (S
            (VP (TO to)
              (VP
                (VP (VB sleep)
                  (PP (IN in)
                    (NP (PRP$ their) (NNS offices))))
                (CC or)
                (VP (VB walk)
                  (NP (NN home))
                  (PP (IN during)
                    (NP (DT the) (NN night)))))))))
    (, ,)
    (NP (NNS officials))
    (VP (VBD said)
      (NP-TMP (NN today)))
    (. .)))
```

NLP Tools (cont'd)

- Boxer:
  - Semantic analysis tool.
  - Takes analysis one step further than parse trees.

"Pierre Vinken, 61 years old , will join the board as a nonexecutive director Nov. 29"



NLP Tools (cont'd)

- WordNet:
  - Lexical database of synonyms
    - i.e. thesaurus
  - Demo:
    - http://wordnetweb.princeton.edu/perl/webwn
  - Useful for questions that aren't resolving
    - e.g. questions that aren't translating well

Project-specific issues

- Immigration Law:
  - Will you allow users to ask their own questions?
    - New immigrants may not know what questions to ask, or even how to ask the questions.
    - On the other end of the spectrum, menu systems can be limiting and frustrating.
  - How will you deliver your answers?
    - Certain questions call for lists, others for text.

Project-specific issues

- Judge Profiler:
  - Key input issues will be related to UI.
  - Processing results is largely a regex task.
    - Do initial tests to detect formatting issues, or language issues.

Project-specific issues

- ROSS (crowdsourced law researcher):
  - UI will also be the crown jewel of this app.
    - User queries → Watson queries:
      - largely a 1-to-1 relationship, with some input sanitization.
    - Displaying the output might entail candidate generation within the passage response.
  - Recruiting approvers (lawyers and non-lawyers) will be important to perfect the experience.

Project-specific issues

- Divorce (relationship? empowerment?) app:
  - To maximize the potential of Watson, user interaction will probably be in natural language.
    - Will have to handle more inputs than just questions, with appropriate responses.
  - Include advice columns as well as legal text, to help provide comfort as well as information.
    - Remember that you're sharing ☺ Be careful not to include anything that conflicts with the documents for any other app!

Project-specific issues

- Sherlock (document recommender app):
  - Examine other recommender systems (i.e. Netflix, Amazon) for language features that they use.
  - You'll want to look into document summarization techniques or sentence extraction to seed the questions that will be used to search for related documents.
    - Use the current browser window to help you focus on the topics for your questions.

General NLP advice

- Important to pre-process your data, to increase the likelihood of getting well-formed responses.
  - e.g. flattening out statute parts into composed sentences.
- Get a representative sample of questions, and make sure the corpus matches the language and content.

General Watson NLP advice

- Maintain current knowledge status, and include that when constructing your Watson query.
- Find approvers, and get them started early!
  - Training needs to happen throughout the development process.
- Consider importing parallel data sources, outside of CanLII.
  - e.g. FAQs on immigration law from government websites, law articles related to marriage & divorce.