Introduction to “This is Watson”

Paper # 10.1147/JRD.2012.2184356

Spencer Carver
Matthew Mercer

Watson

- IBM computer designed to compete on the game show Jeopardy!
- Culmination of 4 years of dedicated research, and over a decade of advancements in algorithms pertaining to natural language parsing (NLP)

Beginnings

- Project proposed in 2004 when Ken Jennings was on his 74 game Jeopardy! winning streak
- Not picked up until 2007, as many at IBM thought that NLP was too difficult to tackle
- Team led by David Ferrucci

PIQUANT

- Originally a project worked on from 1999 to 2005
- Used single ontology and combined parsing and information extraction in order to provide answers to factoid-like questions
- Answers were determined based on categories

PIQUANT (continued)

- However, Jeopardy! problems were much more difficult than PIQUANT could handle (Video clip: 11:15 – 12:45)
- IBM determined to implement WATSON in this manner would require 2000+ different unique categories
- Different approach would be necessary

DeepQA

- DeepQA was a new, extensible software written for Watson
- Pipelined System for determining answers from a given question
- Main stages: Process Question, Generate Hypothesis, Support Hypothesis, Combine Hypotheses, Confidence Ranking

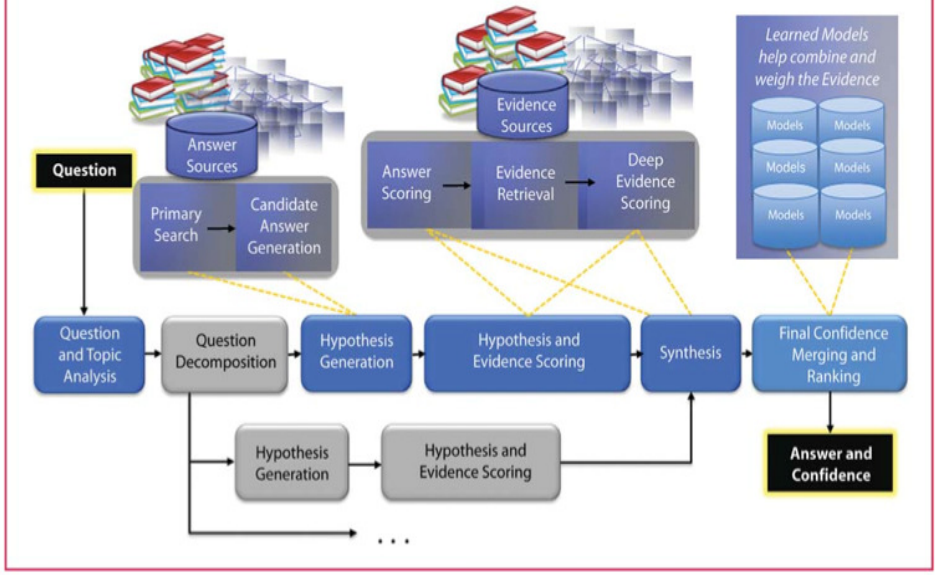

Figure 1

DeepQA architecture.

Question Processing and Generating Hypotheses

- Never assume the question is understood
- Parse the question into possible interpretations
- Run many queries against various sources
- Take each result of these queries as a candidate answer

Supporting and Combining Hypotheses

- Each hypothesis is independently supported
- Evidence is found to determine probability that the hypothesis is valid
- The largest probability is chosen as the correct answer

Confidence Rating

- The hypotheses are rated based on many different metrics, such as popularity and source credibility
- The weights of each metric are learned from past experiments with questions
- If the score for the hypothesis is high enough, Watson will attempt to answer the question

Finding The Right Answer

- When searching for an answer, DeepQA attempts to find answers of a certain type
- Those types are determined from context and are not predefined
- Each type is given a hypothesis and is weighted to determine the chance of it being correct

Finding The Right Answer Fast

- As originally programmed, all the verification algorithms took just over 2 hours to compete
- Utilized the UIMA-AS architecture to distribute the load to 2,880 processors each running asynchronous operations
- Allowed Watson to reduce his average to 3 seconds

Advanced & Special Techniques

- Machine Learning (Video Clip: 41:40 – 44:10)
- Recognition of word play puzzles and puns
- Jeopardy! game interface

Results

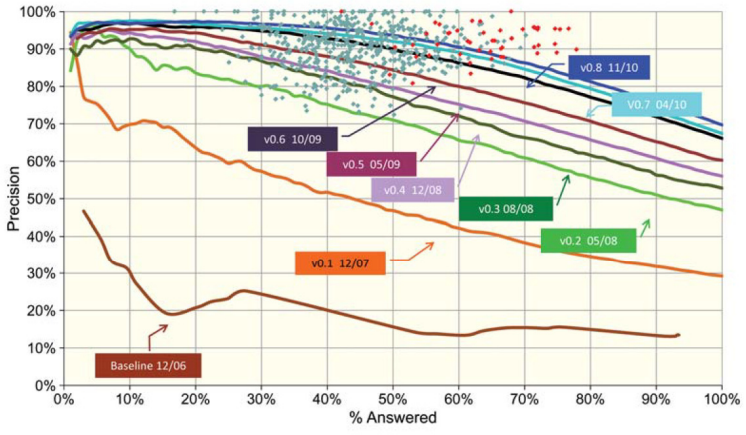

Figure 2

Incremental progress in answering precision on the Jeopardy! challenge: June 2007 to November 2011.

Results (continued)

- The original goal for Watson was to answer 85% of questions correct when buzzing in 70% of the time
- Watson met or exceeded this goal, even hitting 92% accuracy when only buzzing in 40% of the time
- Won 71% of the 55 practice Jeopardy! games

Future Uses

- Differential Diagnosis in Healthcare
- Continued improvements to NLP algorithms

Table 1 DeepQA technology performance on public benchmark sets. (ACE: automatic content extraction; RTE: recognizing textual entailment.)

NLP task	*Evaluation set*	*Project start*	*State of art*	*Watson*
Parsing	Wikipedia** accuracy	84.4	81.1 Charniak parser [19]	88.7
Entity disambiguation	Wikipedia disambiguation F_1	72.5	81.9 Hoffart et al. [42]	92.5
Relation detection	ACE 2004 F_1	45.8	72.1 Zhang et al. [43]	73.2
Textual entailment	RTE-6 2010 F_1	34.6	48.0 PKUTM [44]	48.8