IBM Watson

Berfu Büyüköz - Mert Çıkla
CmpE 561

Boğaziçi University

May 9, 2017

Outline

1. Jeopardy!
2. IBM and QA
3. DeepQA
   - Question Analysis
   - Resource Acquisition
   - Candidate Answer Search & Generation
   - Candidate Ranking with Evidence
   - Structured Data
4. Watson as a Jeopardy player
   - Making Watson Fast
   - In-Game Strategy Optimization
5. Conclusion
6. References

Jeopardy!

- Considered a hard game for humans
- Even harder for any computer system at the time
- Turkish adaptation Riziko!

Category: Types of Financial Aid

Involves money that does not have to be repaid, and generally based on merit.
Scholarship

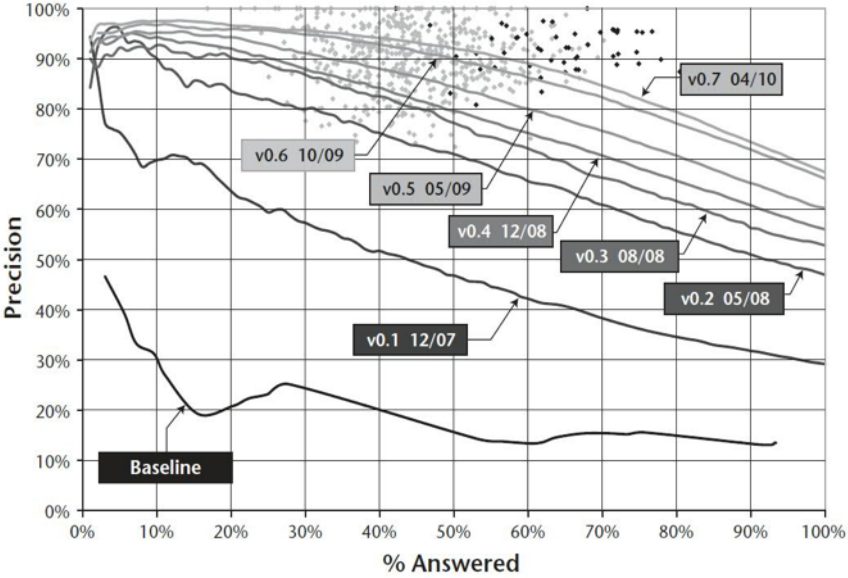
Category: Number Word Play

This number, one of the first 20, uses only one vowel (4 times!)
Seventeen

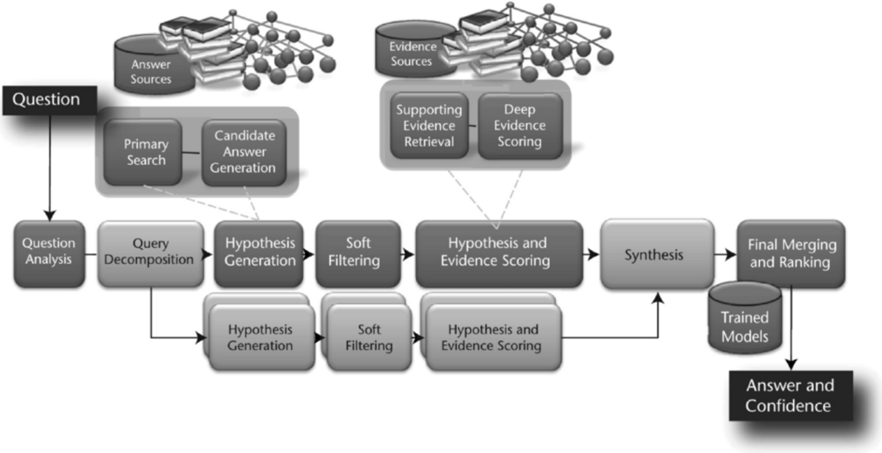
IBM and QA

	% attempted	% correct
PIQUANT	70	16
Winning Players	40-50	85-95
Watson's Goal	70	85

Watson Iterations



DeepQA Pipeline



Question Analysis

- Watson never assumes any component perfectly understands the question
- Keeps collecting evidence from hundreds of algorithms and ranks them
- AdaptWatson is used to develop, test and optimize the algorithms

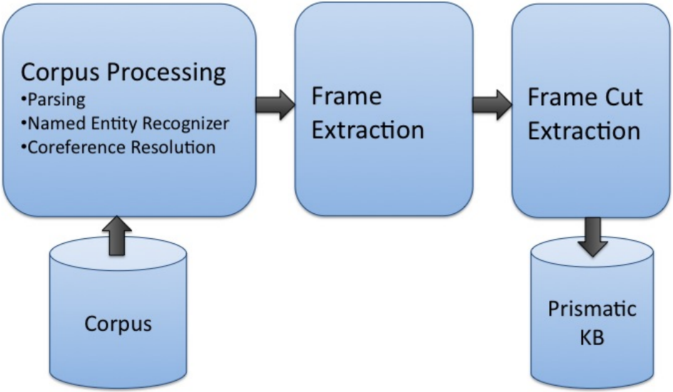
Resource Acquisition

Success of any QA system with limited resources is very dependent on the quality of its data resource.

Wide domain range of Jeopardy makes this even more critical. Watson tackles this in three separate ways

1. Fill the gaps in knowledge space
2. Transform existing knowledge into new information bits
3. Increase existing knowledge by adding new lexical/syntactic variations

Prismatic Knowledge Base



- Employs *is_a* relations derived from large corpora
- Amounts to 2.4% overall increase in precision

Candidate Answer Search & Generation

Finding an exact answer is rare (2 % at best)
The rest is extracted from many algorithms with 3 main strategies

1. Title-in-clue Based Passage Search
2. Passage search in unstructured resources
3. Direct answer lookup & Prismatic

Evidence Scoring

Type Coercion- TyCor

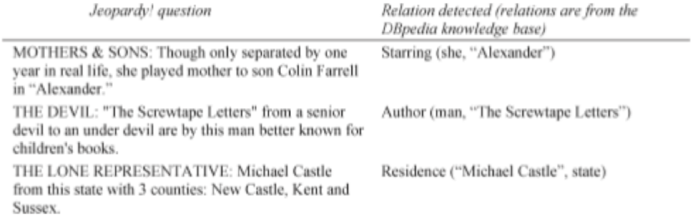
- Performed after candidate generation and before ranking
- Assign scores to candidate answers based on whether it has a specified lexical type.

Who wrote The Hobbit?

“Dan Brown wrote several books and has read The Hobbit.”

“Tolkien, an English author born in the late nineteenth century, wrote The Hobbit.”

Candidate Ranking with Evidence

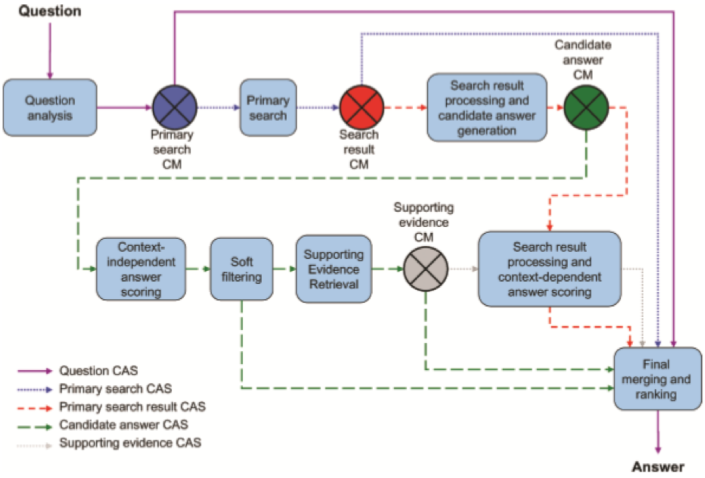


Jeopardy! question	*Relation detected (relations are from the DBpedia knowledge base)*
MOTHERS & SONS: Though only separated by one year in real life, she played mother to son Colin Farrell in "Alexander."	Starring (she, "Alexander")
THE DEVIL: "The Screwtape Letters" from a senior devil to an under devil are by this man better known for children's books.	Author (man, "The Screwtape Letters")
THE LONE REPRESENTATIVE: Michael Castle from this state with 3 counties: New Castle, Kent and Sussex.	Residence ("Michael Castle", state)

1. Large online databases like Wikipedia
2. Large collections of automatically extracted data from unstructured sources
3. Small amount of manually added sources to account for differences between task domain and the source
4. Small amount of manually added formal knowledge targeting most common questions/answers.

Making Watson Fast

From 2 hours to 3 seconds with 2880 processors, preloading to RAM to eliminate disk I/O and preprocessing.


Question
Question analysis
Primary search CM
Primary search
Search result CM
Search result processing and candidate answer generation
Candidate answer CM
Context-independent answer scoring
Soft filtering
Supporting Evidence Retrieval
Supporting evidence CM
Search result processing and context-dependent answer scoring
Final merging and ranking
Answer
Question CAS
Primary search CAS
Primary search result CAS
Candidate answer CAS
Supporting evidence CAS

In-Game Strategy Optimization

- Daily Doubles
- Final Jeopardy
- Square selection
- Attempt or pass decision

Conclusion

Watson beating humans at Jeopardy marks a milestone for QA systems.

Diverse problems encountered during its development helped further research in NLP immensely by introducing new state of the art techniques for these problems.

Thank you for listening

References

1. Ferrucci, David A. "Introduction to “this is watson”." IBM Journal of Research and Development 56.3.4 (2012): 1-1.
2. Lally, Adam, et al. "Question analysis: How Watson reads a clue." IBM Journal of Research and Development 56.3.4 (2012): 2-1.
3. McCord, Michael C., J. William Murdock, and Branimir K. Boguraev. "Deep parsing in Watson." IBM Journal of Research and Development 56.3.4 (2012): 3-1.
4. Chu-Carroll, Jennifer, et al. "Textual resource acquisition and engineering." IBM Journal of Research and Development 56.3.4 (2012): 4-1.
5. Fan, James, et al. "Automatic knowledge extraction from documents." IBM Journal of Research and Development 56.3.4 (2012): 5-1.
6. Chu-Carroll, Jennifer, et al. "Finding needles in the haystack: Search and candidate generation." IBM Journal of Research and Development 56.3.4 (2012): 6-1.
7. Murdock, J. William, et al. "Typing candidate answers using type coercion." IBM Journal of Research and Development 56.3.4 (2012): 7-1.
8. Murdock, J. William, et al. "Textual evidence gathering and analysis." IBM Journal of Research and Development 56.3.4 (2012): 8-1.
9. Wang, Chang, et al. "Relation extraction and scoring in DeepQA." IBM Journal of Research and Development 56.3.4 (2012): 9-1.
10. Kalyanpur, Aditya, et al. "Structured data and inference in DeepQA." IBM Journal of Research and Development 56.3.4 (2012): 10-1.
11. Prager, John M., Eric W. Brown, and Jennifer Chu-Carroll. "Special questions and techniques." IBM Journal of Research and Development 56.3.4 (2012): 11-1.
12. Chu-Carroll, Jennifer, et al. "Identifying implicit relationships." IBM Journal of Research and Development 56.3.4 (2012): 12-1.
13. Kalyanpur, Aditya, et al. "Fact-based question decomposition in DeepQA." IBM Journal of Research and Development 56.3.4 (2012): 13-1.
14. Gondek, D. C., et al. "A framework for merging and ranking of answers in DeepQA." IBM Journal of Research and Development 56.3.4 (2012): 14-1.
15. Epstein, Edward A., et al. "Making watson fast." IBM Journal of Research and Development 56.3.4 (2012): 15-1.
16. Tesauro, Gerry, et al. "Simulation, learning, and optimization techniques in Watson's game strategies." IBM Journal of Research and Development 56.3.4 (2012): 16-1.
17. Lewis, Burn L. "In the game: The interface between Watson and Jeopardy!." IBM Journal of Research and Development 56.3.4 (2012): 17-1.