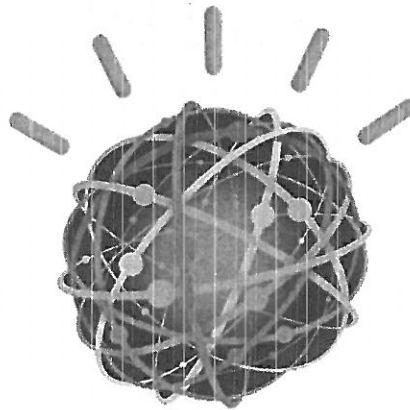


- 5.8 Search Engines 267
- 5.9 Web Usage Mining (Web Analytics) 274
- 5.10 Social Analytics 281

5.1 OPENING VIGNETTE: Machine Versus Men on *Jeopardy!*: The Story of Watson



Can machine beat the best of man in what man is supposed to be the best at? Evidently, yes, and the machine's name is Watson. Watson is an extraordinary computer system (a novel combination of advanced hardware and software) designed to answer questions posed in natural human language. It was developed in 2010 by an IBM Research team as part of a DeepQA project and was named after IBM's first president, Thomas J. Watson.

BACKGROUND

Roughly 3 years ago, IBM Research was looking for a major research challenge to rival the scientific and popular interest of Deep Blue, the computer chess-playing champion, which would also have clear relevance to IBM business interests. The goal was to advance computer science by exploring new ways for computer technology to affect science, business, and society. Accordingly, IBM Research undertook a challenge to build a computer system that could compete at the human champion level in real time on the American TV quiz show, *Jeopardy!* The extent of the challenge included fielding a real-time automatic contestant on the show, capable of listening, understanding, and responding—not merely a laboratory exercise.

COMPETING AGAINST THE BEST

In 2011, as a test of its abilities, Watson competed on the quiz show *Jeopardy!*, which was the first ever human-versus-machine matchup for the show. In a two-game, combined-point match (broadcast in three *Jeopardy!* episodes during February 14–16), Watson beat Brad Rutter, the biggest all-time money winner on *Jeopardy!*, and Ken Jennings, the record holder for the longest championship streak (75 days). In these episodes, Watson consistently outperformed its human opponents on the game's signaling device, but had trouble responding to a few categories, notably those having short clues containing only a few words. Watson had access to 200 million pages of structured and unstructured content consuming four terabytes of disk storage. During the game Watson was not connected to the Internet.

Meeting the *Jeopardy!* Challenge required advancing and incorporating a variety of QA technologies (text mining and natural language processing) including parsing, question classification, question decomposition, automatic source acquisition and evaluation, entity and relation detection, logical form generation, and knowledge representation and reasoning. Winning at *Jeopardy!* required accurately computing confidence in your answers. The questions and content are ambiguous and noisy and none of the individual algorithms are perfect. Therefore, each component must produce a confidence in its output, and individual component confidences must be combined to compute the overall confidence of the final answer. The final confidence is used to determine whether the computer system should risk choosing to answer at all. In *Jeopardy!* parlance, this confidence is used to determine whether the computer will “ring in” or “buzz in” for a question. The confidence must be computed during the time the question is read and before the opportunity to buzz in. This is roughly between 1 and 6 seconds with an average around 3 seconds.

HOW DOES WATSON DO IT?

The system behind Watson, which is called DeepQA, is a massively parallel, text mining-focused, probabilistic evidence-based computational architecture. For the *Jeopardy!* challenge, Watson used more than 100 different techniques for analyzing natural language, identifying sources, finding and generating hypotheses, finding and scoring evidence, and merging and ranking hypotheses. What is far more important than any particular technique that they used was how they combine them in DeepQA such that overlapping approaches can bring their strengths to bear and contribute to improvements in accuracy, confidence, and speed.

DeepQA is an architecture with an accompanying methodology, which is not specific to the *Jeopardy!* challenge. The overarching principles in DeepQA are massive parallelism, many experts, pervasive confidence estimation, and integration of the-latest-and-greatest in text analytics.

- **Massive parallelism:** Exploit massive parallelism in the consideration of multiple interpretations and hypotheses.
- **Many experts:** Facilitate the integration, application, and contextual evaluation of a wide range of loosely coupled probabilistic question and content analytics.
- **Pervasive confidence estimation:** No component commits to an answer; all components produce features and associated confidences, scoring different question and content interpretations. An underlying confidence-processing substrate learns how to stack and combine the scores.
- **Integrate shallow and deep knowledge:** Balance the use of strict semantics and shallow semantics, leveraging many loosely formed ontologies.

Figure 5.1 illustrates the DeepQA architecture at a very high level. More technical details about the various architectural components and their specific roles and capabilities can be found in Ferrucci et al. (2010).

CONCLUSION

The *Jeopardy!* challenge helped IBM address requirements that led to the design of the DeepQA architecture and the implementation of Watson. After 3 years of intense research and development by a core team of about 20 researchers, Watson is performing at human expert levels in terms of precision, confidence, and speed at the *Jeopardy!* quiz show.

IBM claims to have developed many computational and linguistic algorithms to address different kinds of issues and requirements in QA. Even though the internals of these algorithms are not known, it is imperative that they made the most out of

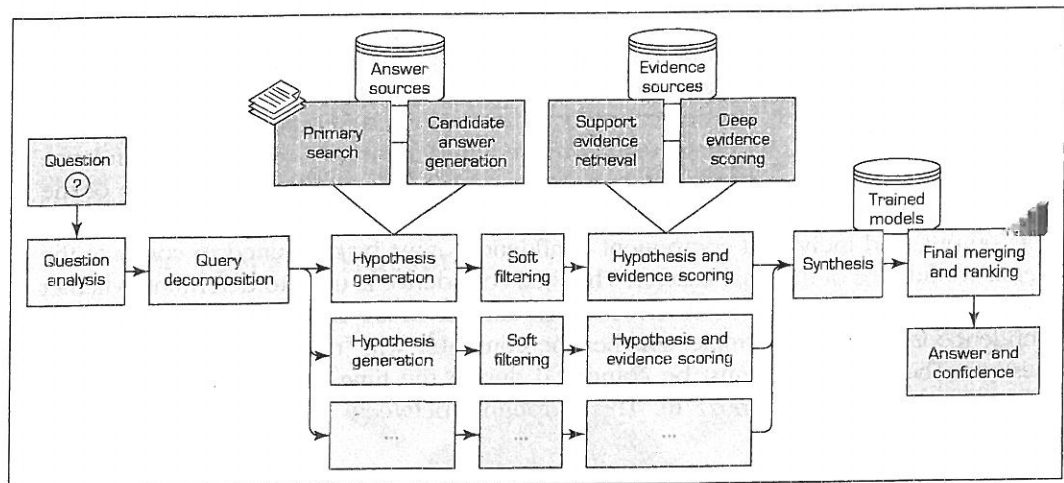


FIGURE 5.1 A High-Level Depiction of DeepQA Architecture.

text analytics and text mining. Now IBM is working on a version of Watson to take on surmountable problems in healthcare and medicine (Feldman et al., 2012).

QUESTIONS FOR THE OPENING VIGNETTE

1. What is Watson? What is special about it?
2. What technologies were used in building Watson (both hardware and software)?
3. What are the innovative characteristics of DeepQA architecture that made Watson superior?
4. Why did IBM spend all that time and money to build Watson? Where is the ROI?

WHAT WE CAN LEARN FROM THIS VIGNETTE

It is safe to say that computer technology, both on the hardware and software fronts, is advancing faster than anything else in the last 50-plus years. Things that were too big, too complex, impossible to solve are now well within the reach of information technology. One of the enabling technologies is perhaps text analytics/text mining. We created databases to structure the data so that it can be processed by computers. Text, on the other hand, has always been meant for humans to process. Can machines do the things that require human creativity and intelligence, and were not originally designed for machines? Evidently, yes! Watson is a great example of the distance that we have traveled in addressing the impossible. Computers are now intelligent enough to take on men at what we think men are the best at. Understanding the question that was posed in spoken human language, processing and digesting it, searching for an answer, and replying within a few seconds was something that we could not have imagined possible before Watson actually did it. In this chapter, you will learn the tools and techniques embedded in Watson and many other smart machines to create miracles in tackling problems that were once believed impossible to solve.

Sources: D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefel, and C. Welty, "Building Watson: An Overview of the DeepQA Project," *AI Magazine*, Vol. 31 No. 3, 2010; DeepQA, DeepQA Project: FAQ, IBM Corporation, 2011, research.ibm.com/deepqa/faq.shtml (accessed January 2013); and S. Feldman, J. Hanover, C. Burghard, and D. Schubmehl, "Unlocking the Power of Unstructured Data," IBM white paper, 2012, www-01.ibm.com/software/ebusiness/jstart/downloads/unlockingUnstructuredData.pdf (accessed February 2013).