Cognitive Computing and Big Data Analytics
By Judith Hurwitz, Marcia Kaufman and Adrian Bowles
Copyright © 2015 by John Wiley & Sons, Inc.

CHAPTER

9

IBM's Watson as a Cognitive System

One of the best ways to understand the potential for cognitive computing is to take a look at one of the early implementations of a cognitive system. IBM developed Watson as one of its new foundational offerings intended to help customers build a different type of system based on the ingestion of new content. IBM's design focus for Watson was to create solutions based on aggregating data leveraging techniques ranging from machine learning to Natural Language Processing (NLP) and advanced analytics. Watson solutions include a set of foundational services combined with industry-focused best practices and data. The accuracy of results from a cognitive system continuously improves through an iterative training process that combines the knowledge of subject matter experts with a corpus of domain specific data. One of the important capabilities that allows for this machine/human interaction is the ability to leverage NLP to understand the context of a combination of a variety of unstructured and structured data sources. In addition, a cognitive system is not constrained to applications that are deterministic in nature, but can manage probabilistic systems that change and evolve as they are used.

Watson Defined

Watson is a cognitive system that combines capabilities in NLP, analytics, and machine learning techniques. Watson gains insights and gets smarter with each

user interaction and each time that new information is ingested. By combining NLP, dynamic learning, and hypothesis generation and evaluation, Watson is intended to help professionals create hypotheses from data, accelerate findings, and determine the availability of supporting evidence to solve problems. IBM views Watson as a way to improve business outcomes by enabling humans to interact with machines in a natural way.

Individuals have become accustomed to leveraging sophisticated search engines or database query systems to discover information to support decision making. Watson, which also facilitates data-driven search, takes a different approach that is discussed in detail in this chapter. In essence, Watson leverages machine learning, DeepQA, and advanced analytics. IBM Watson's DeepQA architecture, as illustrated in Figure 9-1, is described in this chapter.

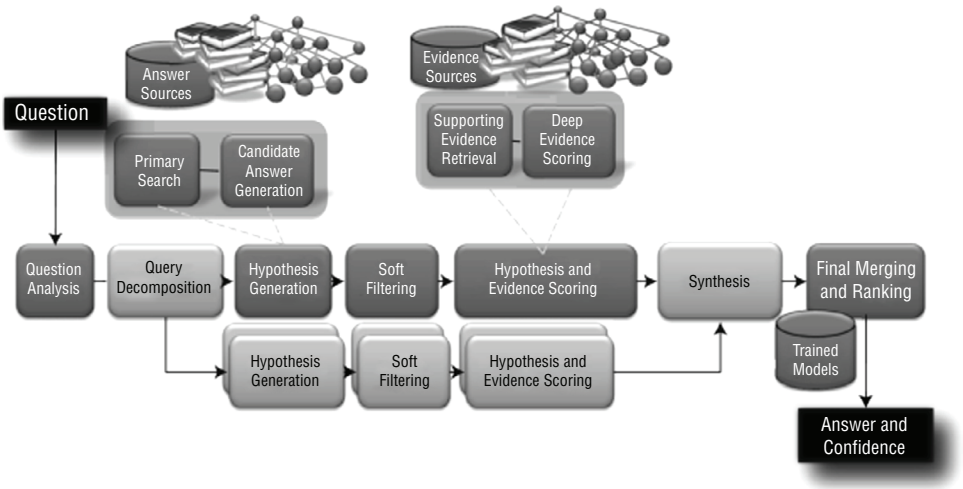

Figure 9-1: IBM Watson DeepQA Architecture

Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

How Watson Is Different from Other Search Engines

One way to understand this unique process is to consider how Watson, as a cognitive computing system, differs from search engines. With a search engine, you enter key words and get results on a topic based on an appropriate ranking. You might also ask a specific question and receive ranked results; however, you will not be able to create a dialogue to continue to refine the results. A typical search engine uses algorithms to rank results based on relevance to the keywords. Secondary rankings may deliver results based on factors such as price or consumer reviews. At this point, humans interact with the list of results and assess which answers or links best fit the question being asked.

With Watson, the individual gets a directed result—either an answer to the question or a follow-up question to help clarify the user's intent. Therefore, the machine is intended to act more like a human expert. For example, the user might ask Watson, "What is the best retirement program for me?" or "What is the best way to lose weight?" If Watson has enough data and enough contextual knowledge related to the subject, the system can understand the language behind the question. This deep level of understanding is driven through the use of statistical analysis and algorithms for developing predictive models. Watson does not simply look for keywords as a search engine would. In addition, Watson, leveraging NLP techniques, has the capability to break a question into subcomponents and evaluate each component for possible answers and solutions. This capability of receiving meaningful, accurate, and timely answers to a direct question is the fundamental difference between most search engines and the question-answering process of cognitive systems.

Advancing Research with a "Grand Challenge"

Computer scientists have often used games as a way to advance their research agenda and publicly demonstrate innovative computer technology. In keeping with this tradition, IBM has a long history of incorporating games into "grand challenges" for its research teams. Two of IBM's highly publicized "grand challenges" involved the game of checkers in the 1950s and the game of chess in the late 1990s. One of the early pioneers of Artificial Intelligence (AI) programmed an IBM 701 computer to play checkers, and beat one of the top U.S. checkers champions. At the time, this feat was a stunning example of the capabilities of computers. More than 30 years later, IBM's Deep Blue computer was the first to beat a world chess champion. The purpose of a "grand challenge" is to take a theoretical concept and prove it can be done.

IBM's success with its chess grand challenge was based on computers beating humans in the world of mathematics. Researchers at IBM thought that the next challenge should explore and advance the capabilities of computers in human natural language and knowledge. In 2006, IBM outlined a grand challenge that could help transform the way businesses make decisions. One of IBM's researchers suggested that the company build a computer that could compete with human champions and win at *Jeopardy!* The initial focus was to determine if a system could compete against humans by answering questions across a diverse range of topical areas. The biggest issue that IBM researchers faced in succeeding with the grand challenge was establishing the right balance between speed of processing and accuracy of results. Although the goal of the grand challenge was to beat humans at the game of *Jeopardy!* IBM hoped to be able to use the Jeopardy! challenge as a way to begin investigating the potential to create a cognitive systems that could support complex industries.

Preparing Watson for *Jeopardy!*

IBM brought together an internal team of expert scientists and researchers in fields ranging from machine learning to mathematics, High Performance Computing (HPC), NLP, and knowledge representation to translate the grand challenge into a platform. In order to win at *Jeopardy!*, the team would need to build a system that could answer questions asked in human language faster and more accurately than the top ranked human competitors. IBM determined that Watson would need to answer approximately 70 percent of questions and get the right answer more than 80 percent of the time. Further, this level of accuracy would need to be accomplished in 3 seconds or less for each question.

To accomplish these goals, Watson was designed as a Question-Answering system that uses continuously acquired knowledge to determine answers to questions and confidence scores associated with those questions. Watson understands the context of a sentence by deconstructing each element of the sentence, comparing those elements against previously ingested information, and making inferences as to meaning.

The complexity of the *Jeopardy!* challenge was based on the diversity of question types and the broad base of subject areas included in the game. In addition, there is always one right answer or response to a question in *Jeopardy!* You can't respond to a question with a request for additional information. Contestants need to discern the question based on a set of clues. Clues could be technical information or they could be puns, puzzles, or cultural references. To solve the clue at fast enough speeds to win the game, Watson needed to understand many aspects of language that humans understand instinctively. Humans have a natural ability to understand inference, context, and constraints of time and space.

To ensure that responses are highly accurate, Watson simultaneously generates many hypotheses (potential answers) in a parallel computing environment. These hypotheses need to be generated in a way that casts a broad enough net so that the right answer is among the selections, but not so broad that large numbers of incorrect hypotheses interfere with the overall efficiency of the process. Sophisticated algorithms rank and determine a confidence level for each hypothesis. Advances in natural language processing technologies helped to make this approach a reality. IBM built an architecture that would support the use of machine learning to do extensive experimentation to continuously advance Watson's cognitive capabilities.

The need for computational speed led IBM to use extremely fast and powerful hardware. On the night of the televised *Jeopardy!* competition, Watson included a combination of servers, storage, memory, and networking equipment that placed it in a supercomputer class. Watson included 90 IBM Power 750 servers, each with four processors for a maximum of 32 logical cores per processor. This means that there was a total of 2,880 IBM Power7 Processor Cores. It was the

power of the 2,880 cores that allowed Watson to meet the 3-second requirement for delivering an answer to a question. In addition, Watson was designed to store its entire knowledge base in random access memory (RAM) instead of on disk to further speed up processing speeds and deliver fast results. Extremely fast networking technology was included to help move a lot of data between compute nodes at fast speeds.

Preparing Watson for Commercial Applications

The question-answering process for the game of *Jeopardy!* is different than what you would typically expect in commercial applications. Instead of providing one right answer to a question in *Jeopardy!*, commercial applications require more complex, multidimensional answers. Commercial applications built for industries such as healthcare and finance need to support an ongoing dialogue between humans and machines that would help to drill down to the most meaningful set of responses. In addition, Watson would be expected to ask for more information when needed to help the business users get the most useful and accurate response.

The difference between a typical *Jeopardy!* question and a sample question for a commercial healthcare application is illustrated in Table 9-1. The question from *Jeopardy!* includes a subject domain and a statement about an entity or concept. The entity or concept is not identified in the statement. The subject domain in this question is "delicacies" and the unidentified entity is "pig." In commercial applications for Watson, such as the Watson Discovery Advisor, it is unlikely that there will be just one correct answer. For example, the question shown in the following table asks for a treatment plan for a patient. The intention is for the physician to engage in a collaborative dialogue with Watson.

Table 9-1: Answering a *Jeopardy!* Question Compared to Answering a Watson Discovery Advisor Question

STANDARD *JEOPARDY*! QUESTION AND ANSWER	STANDARD QUESTION AND ANSWER FOR WATSON DISCOVERY ADVISOR
Question: DELICACIES: Star chef Mario Batali lays on the lardo, which comes from the back of this animal's neck.	Question: An oncologist is reviewing treatment options with a cancer patient and asks Watson, "What is the recommended treatment for patient X?"
Answer: The answer to the question is "pig."	Answer: The answer to the question is multifaceted and is provided as an ongoing dialogue with the oncologist. The answer may include recommendations for additional tests and provide options for various treatments.

Advanced machine learning techniques are used to train Watson to provide correct answers to many types of questions, including those illustrated in Table 9-1. Watson arrives at the answer by considering many possible responses to the question based on its body of knowledge (corpus). In addition, Watson looks at the context of the question from many different approaches and considers different interpretations and definitions for words and phrases. Each of the possible answers is given a confidence value by Watson. Watson provided the single answer "pig" to the sample *Jeopardy!* question because this was the answer with the highest confidence level. In comparison, Watson may provide several alternative answers to the question about treatment options and show the confidence level for each answer.

IBM is applying many of the technology advancements that helped Watson win at *Jeopardy!* to its commercial applications of cognitive systems. These systems use evidenced-based learning to enable organizations to train systems to get smarter with each new interaction. Training is an important aspect of implementing a Watson system in a commercial environment. The training data includes question and answer pairs on how things are said in that specific industry. Watson can also be trained for a new industry by ingesting resources such as an ontology. For example, in a Watson application for a hospital, the training might include ingesting a deep ontology or coding system specific to medical diagnostic testing or treatments for specific diseases. Ontologies provide a mechanism for determining context by clarifying and defining terminology and creating accurate mappings between resources from different systems. In addition, standards-based guidelines on how to treat specific diseases would be included. Additional training may be based on the clinical expertise of highly knowledgeable and experienced clinicians. Companies can use these cognitive systems to answer new types of questions, make more accurate predictions, and optimize business outcomes.

Watson's Software Architecture

The design structure of Watson includes software architecture for building Question-Answering systems and a methodology to research, develop, and integrate algorithmic techniques into the system. Although speed and power are critical elements for Watson, the design team initially focused on achieving accuracy and confidence. Without these characteristics, the speed would be meaningless. Therefore, a key design element includes algorithms for assessing and increasing accuracy. The Natural Language Processing technologies incorporated into the Watson architecture—known as DeepQA—include the following:

- Question parsing and classification
- Question decomposition

- Automatic source acquisition and evaluation
- Entity and relation detection
- Logical form generation
- Knowledge representation and reasoning

The DeepQA software architecture is built according to Unstructured Information Management Architecture (UIMA) standards. UIMA was initially created by IBM and then open-sourced to the Apache Software Foundation. It was chosen as the framework for the hundreds of analytic components in DeepQA because of its capability to support the extreme speed, scalability, and accuracy required across a large number of distributed machines. Through experimentation, IBM improved the accuracy of the DeepQA algorithms and, consequently, confidence in Watson's results. The following were DeepQA's core design principles:

- **Massive Parallelism**—A large number of computer processes work in parallel to optimize processing speed and overall performance. Using this technique enables Watson to analyze vast sources of information and evaluate different interpretations and hypotheses at extremely fast speeds.
- **Integration of probabilistic question and content analytics**—Algorithms and models are developed using machine learning to provide correct answers that assume deep levels of expertise across multiple domains. The corpus provides a base of knowledge and the analytics estimate, and understands the relationships and patterns in the information.
- **Confidence estimation**—The architecture is designed in such a way that there are multiple interpretations to a question. There is never a commitment to a single answer. The approach of continually scoring different answers with a confidence level is key to Watson's accuracy. The technology analyzes and combines scores of different interpretations to understand which interpretation is most relevant.
- **Integration of shallow and deep knowledge**—Shallow knowledge is procedural in nature and does not support the ability to make connections between different elements of a particular subject area. You can use shallow knowledge to get the answer to certain types of questions, but there are many limitations. To go deeper than a literal or superficial understanding of question and response, you need to make associations and inferences. To achieve this level of sophistication, you need deep knowledge, which is about understanding the central foundational concepts of a particular subject area—such as investment banking or medical oncology. With deep knowledge you can make complex connections and associations to those central concepts.

The methodology for the development and integration of core algorithmic techniques is called AdaptWatson. The methodology creates core algorithms, measures the results, and then comes up with new ideas. AdaptWatson quickly manages the research, development, integration, and evaluation of the core algorithmic components. The algorithmic components have many roles including:

- Understanding questions
- Creating the confidence level of the answer
- Evaluating and ranking the results
- Analyzing natural language
- Identifying sources
- Finding and generating hypotheses
- Scoring evidence and answers
- Merging and ranking hypotheses

To determine relationships and inferences, Watson uses machine learning and linear regression to rank data based on relevance.

The Components of DeepQA Architecture

The essential components of the Watson DeepQA architecture include a pipeline process flow that begins with a question and concludes with an answer and confidence level (refer to Figure 9-1). The various answer sources come up with alternative responses, and then each response is evaluated and ranked as to its likelihood of being a correct response. There is an iterative process that needs to take place in seconds but will allow for evidence to be collected and analyzed before the best answer is determined. The components in DeepQA are implemented as UIMA annotators. These annotators are software components that analyze text to create assertions (or annotations) about that text. At each stage, there is a role for an UIMA annotator to help move the process forward. Watson has hundreds of UIMA annotators. The different types of capabilities that need to take place within the pipeline are as follows:

- **Question analysis**—Each question is parsed to extract major features and begin the process of understanding what is asked by the question. This analysis determines how the question will be processed by the system.
- **Primary search**—Content is retrieved from the evidence and answer sources.
- **Candidate answer generation**—Various hypotheses (candidate answers) are extracted from the content. Each potential answer is considered as a candidate for the correct answer. NLP interprets and analyzes the text

search results. Answer and evidence sources are examined to provide insight into how to answer the question. Hypotheses or candidate answers are generated by this analysis. Each hypothesis is considered and reviewed independently.

- **Shallow answer scoring**—The various candidates for the answer are scored across many dimensions such as geospatial similarity.
- **Soft filtering**—After each candidate answer is scored, the soft filtering process scores and selects approximately the top 20 percent of the scored candidates for additional analysis.
- **Supporting evidence search**—Additional evidence is researched and applied to the analysis of the top candidates. NLP analysis is performed on the additional supporting evidence. Various hypotheses are tested.
- **Deep Evidence Scoring**—Each piece of evidence is evaluated, using multiple algorithms, to determine to what degree the evidence supports that the candidate answer is correct.
- **Final merging and ranking**—All the evidence for each candidate answer is combined. Ranks are assigned and confidence scores are computed.

The process flow highlighted depends on the answer and evidence sources as well as the models (refer to Figure 9-1). The major components of this architecture are listed next and will be described in more detail in the remainder of this chapter:

- Building the Watson corpus
- Question analysis
- Hypotheses generation
- Scoring and confidence estimation

Building the Watson Corpus: Answer and Evidence Sources

The Watson corpus provides the base of knowledge used by the system to answer questions and provide responses to queries. The corpus needs to provide a broad base of information as reference sources without adding unnecessary information that might slow down performance. IBM looked at the domain of questions that could be included in *Jeopardy!* and the data sources that would be needed to answer those questions. The hardware was scaled up to provide the computational power required to answer approximately 70 percent of the questions and get the right answer approximately 80 percent of the time. The corpus was developed to provide access to vast amounts of information on a broad range of topics. As Watson was leveraged to meet requirements of commercial applications in areas such as healthcare and financial services, the corpus and

ontologies would also need to be developed to provide more domain-specific information. Therefore, IBM developed an approach that would construct the Watson corpus with relevant sources of the right size and breadth to deliver accurate and fast responses. This approach includes three phases:

- **Source acquisition**—Identify the right set of resources for the specific task.
- **Source transformation**—Optimize the format of the textual information for efficient search.
- **Source expansion and updates**—Expansion algorithms are used to determine which additional information would do the best job of filling in gaps and adding nuance to the information sources in the Watson corpus.

Next, each of the three phases is described in more detail.

Source Acquisition

The appropriate sources for building the Watson corpus will vary based on how Watson will be used. One of the first steps is to analyze the subject matter requirements to understand the types of questions that will be asked. Given the broad domain of knowledge required for *Jeopardy!*, the sources for Watson include a diverse collection of texts including encyclopedias, Wikipedia, dictionaries, historical documents, textbooks, news articles, music databases, and literature. Information sources may also include subject-specific databases, ontologies, and taxonomies. The goal is to collect a rich base of knowledge across multiple domains including science, history, literature, culture, politics, and governments. Building the Watson corpus for commercial applications in areas such as healthcare or finance is different than for *Jeopardy*! For example, building the oncology reference corpus requires ingesting vast amounts of information sources on relevant scientific research, medical textbooks, and journal articles.

The majority of the information sources are unstructured documents in various formats such as XML, PDF, DOCX, or any markup language. These documents need to be ingested into Watson. The system is designed to create indexes for the documents and store them in a distributed filesystem. The Watson instance has access to that shared filesystem. The Watson corpus provides both the answer sources and evidence sources. Answer sources provide the primary search and candidate answer generation (selection of possible answers). Evidence sources provide answer scoring, evidence retrieval, and deep evidence scoring.

Source Transformation

Textual information sources come in a variety of formats. For example, documents from an encyclopedia are typically title-oriented, meaning that the titles for the documents identify the subject covered in the piece. Other documents

such as news articles are likely to include a title that indicates a point of view (identified as nontitle-oriented or opinion-labeled) and may not be a clear indication of the subject matter in the piece. Search algorithms typically do a better job of locating the information in title-oriented documents. Therefore, some nontitle-oriented articles are transformed to help improve the likelihood that the relationship between the content and potential answers can be easily identified.

Source Expansion and Updates

How do you decide on the right amount of content for the Watson corpus? There needs to be enough information so that Watson can identify patterns and make associations between various elements of information. IBM determined that many of the primary information sources such as encyclopedias and dictionaries provided a good base of knowledge but left many gaps. To fill these gaps, the Watson team developed algorithms that would search the web for additional information with the right context to amplify information in the base or seed documents. These algorithms are designed to score each element of new information relative to the original seed document and include only the new information that appears most relevant.

The Watson corpus also needs to be continuously fine-tuned and updated to ensure accuracy of results. For example, there are approximately 5,000 new articles each week on cancer. Therefore, the Watson corpus for an oncology application needs to be updated constantly with new and relevant information, or it would quickly be out of date. The mechanics of doing incremental ingestion are such that large amounts of documents need to be accessed and continuously ingested into Watson without bringing down the system. In addition, the quality of the ingested information needs to be monitored to eliminate the possibility of corrupting the corpus with erroneous information that could lead to bad answers. For example, consider the question, "What is the best way to lose weight?" There are so many different points of view on this subject. Do you reduce carbohydrates, eliminate sugar, reduce fats, or increase exercise? Is the most recent journal article given more importance, or is the quality and value of the information based on other rating factors such as author expertise?

The process of fine-tuning the corpus to win at a game where there is always a right answer required continuous evaluation to assess Watson's accuracy in relationship to the requirement for extreme speed. IBM used algorithms to test and refine the new resources that should be added to the corpus to increase Watson's accuracy without increasing latency. Technology developed by IBM to increase Watson's speed and accuracy has been used in its commercial applications such as Watson Engagement Advisor and Watson Discovery Advisor.

Question Analysis

Question Analysis ensures that Watson learns what question is asked and determines how the question is processed by the system. The foundation of the Question Analysis process is based on NLP technology, with a focus on parsing, semantic analysis, and question classification. All these techniques are brought together to enable Watson to understand the type and nature of the questions and to detect relationships between entities in the questions. For example, Watson needs to recognize nouns, pronouns, verbs, and other elements of the sentence to understand what the answer should look like. One reason why the *Jeopardy!* challenge helped to advance IBM's research in NLP is that the domain knowledge required to excel at the game is so diverse. In addition, *Jeopardy!* requires an understanding of many different types of questions, including the capability to recognize humor, puns, and metaphors. IBM worked for many years to refine the algorithms used in Watson's Question Analysis.

Question Analysis requires advanced parsing of the questions from both a syntactic and semantic perspective to extract a logical form. In connection with the parsing, syntactic roles are identified and labeled using algorithms that identify subject, object, and other components of the sentence. In addition, semantic parsing can identify the meaning of phrases and the overall question. The results of the parsing helps Watson to learn what information to search for in the corpus. This is where the associations and pattern matching capabilities become important. Questions are analyzed by identifying the patterns based on the data structures from parsing and semantic analysis. Patterns of words in the text can predict other aspects of the meaning of the content. You need a large enough database of the different types of questions to identify the patterns and recognize the similarities across different logical forms.

Four key elements of the question need to be detected for successful Question Analysis:

- **Focus**—The focus is the part of the question that represents the answer. In order to be able to answer correctly, you need to understand the focus of the question. Determining the focus depends on recognizing the patterns of focus types. For example, one common pattern consists of a noun phrase with a determiner "this" or "these." The following *Jeopardy!* clue illustrates this pattern. "THEATRE: A new play based on this Sir Arthur Conan Doyle canine classic opened on the London stage in 2007." The focus in the clue is "this Sir Arthur Conan Doyle canine classic." The parser needs to connect "this" to the headword "classic." The parser needs to be able to tell the difference between a noun-phrase question and a verb phrase.
- **LAT (Lexical Answer Type)**—Watson uses the LAT to help figure out what type of answer is required. For example, is Watson looking for the name of a film, city, or person?

- **Question Classification**—Watson uses Question Classification to determine the type of question it needs to answer. For example, is the question fact-based, or is it a puzzle, or perhaps a pun? Understanding the question type is important so that Watson can select the right approach for answering the question.
- **QSection**—These are fragments of questions that require a unique approach to find the answer. QSection can identify lexical constraints on the answer (for example, the answer must be only three words) and to decompose a question into multiple subquestions.

Slot Grammar Parser and Components for Semantic Analysis

Watson uses a series of deep parsing and semantic analysis components to provide the linguistic analysis of the questions and reference content. The Slot Grammar (SG) parser builds a tree that maps out the logical and grammatical structure. There are SGs for many languages including English, French, Spanish, and Italian. The parser used in Watson is the English Slot Grammar (ESG). ("IBM Research Report: Using Slot Grammar," Michael C. McCord, 2010.) The parser was enhanced for Watson according to the specific requirements of the *Jeopardy!* game. The role of the parser is to break up a sentence into its semantic phrases of a sentence. These semantic roles or phrases are called slots. In addition, the term slots can also refer to the names for argument positions for predicates that represent word senses. Some examples of slots are shown in Table 9-2.

Table 9-2: Slots—Naming Syntactic Roles or Phrases

subj	subject
obj	direct object
iobj	indirect object
comp	predicate complement
objprep	object of preposition
ndet	noun phrase (NP) determiner

To derive the meaning of questions, Watson needs a way to recognize the similarities and differences across many different syntactical patterns. It is quite common for the same thought or action to be expressed in slightly different ways. For example, Figure 9-2 shows two sentences with different syntactic components that share the same meaning. Watson uses the SG parser to recognize the subject, object, indirect object, and other elements of the sentence. In sentence (A), Emily fills the subject slot for the verb "gave" and Jason fills the indirect object slot. Each slot represents a syntactic role within the sentence. In sentence (B), Emily still fills the subject slot for the verb "gave." However,

in this alternative construct of the sentence, "to Jason" fills the indirect object slot. In other words, the indirect object slot is filled by either the noun phrase "Jason" in sentence A or by the prepositional phrase "to Jason" in sentence B. The syntactic component for the SG needs to understand that these two alternative syntactic examples both have the same meaning. In addition, the SG parse trees need to show both a surface syntactic structure and a deep logical structure. Watson then ranks the various parse trees based on a parse scoring system and selects the parse with the highest ranking.

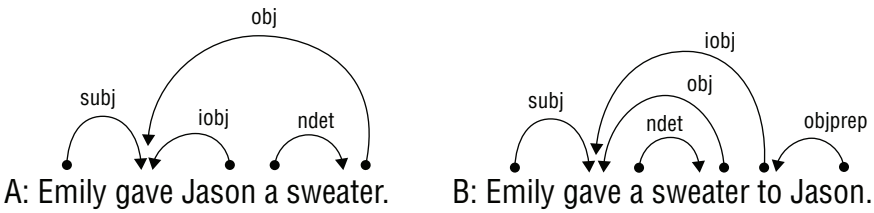

Figure 9-2: Parsing two sentences using English Slot Grammar

In addition to the ESG, Watson uses several other components for parsing and semantic analysis:

- The **Predicate-Argument Structure builder** is used to simplify the ESG tree by mapping small variations in syntax to common forms. It is built on top of ESG to support more advanced analytics.
- The **Named Entity Recognizer (NER)** looks for names, quantities, and locations and determines which terms in the phrase are proper nouns that reference people or organizations.
- The **co-reference resolution component** connects referring expressions to their correct subjects and determines the entities to which pronouns relate.
- The **relation extraction component** looks for semantic relationships in the text. This is important if different terms have a similar meaning and is helpful in mapping the relationship between nouns or entities in the question or clue.

Question Classification

Question Classification is an important element of the Question Analysis process because it helps to identify what type of question is being asked. This process was developed to improve Watson's capability to understand the many different types of clues in *Jeopardy!* You can characterize the clues in *Jeopardy!* by topic, level of difficulty, grammatical construction, answer type, and method to solve the clue. Characterizing the clues based on the method

used to answer the question offered the greatest success with developing Question Classification algorithms. Three of the various methods used to find the right answer follow:

- Answer based on factual information.
- Find the answer by decomposing the clue.
- Find the answer by completing a puzzle.

Identifying the question type will trigger different models and strategies in later processing steps. Watson also uses Relation Detection during the Question Analysis process to evaluate the relationships in the question. One of Watson's greatest strengths is in the way it analyzes the question in great depth, including recognizing nuances and searching across the corpus for different possible answers. (See Table 9-3.)

Table 9-3: Answering Different Types of *Jeopardy!* Clues

TYPE OF CLUE	EXAMPLE	HOW YOU ANSWER THE CLUE
You need to know the facts.	HEAD NORTH: Two states you could be re-entering if you are crossing Florida's northern border. Answer: Georgia and Alabama	You answer the question based on factual information about one or more entities. Understand what is being asked and which elements of the clue will help you get the answer.
You need to decompose the clue.	DIPLOMATIC RELATIONS: Of the four countries in the world that the United States does not have diplomatic relations with, the one that's farthest North. Answer: North Korea	One subclue is nested in the outer clue. After you replace the subclue with its answer, it become easier to answer the outer clue. In this example: The inner subclue is "the four countries in the world that the United States does not have diplomatic relations with." The answer to the subclue is Bhutan, Cuba, Iran, and North Korea. After replacing the subclue with the answer, the new question reads as follows: Of Bhutan, Cuba, Iran, and North Korea, the one that's farthest North.
You need to solve a puzzle.	BEFORE and AFTER: 13th Century Venetian traveler who's a Ralph Lauren short sleeve top with a collar. Answer: Marco Polo	Two subclues have answers that overlap.

Why is it important to understand what is asked in the question? Watson needs to learn based on patterns and associations in question and answer resources. The system does not actually understand concepts that would be

simple for a child to master. For example, a child can learn that two different types of barking creatures are both dogs even though one is a dalmatian and the other is a golden retriever. Machine learning will help Watson to analyze information in many different ways to figure out that the dalmatian and the golden retriever are both dogs. Alternatively, you could ingest Watson with thousands of Question-Answer combinations, but without machine learning Watson would not be able to answer questions that deviated in any way from the original set. Watson needs to learn to answer new types of questions correctly.

Hypothesis Generation

How does Watson find the right answer to a question? Watson's key to success with the Question Analysis process is based on the large number of candidate answers that are considered. Hypothesis Generation (Figure 9-3) can identify various hypotheses to answer a question with the expectation that one of them will be the right answer. Although the right answer needs to be among the candidate answers, you don't want there to be too much noise in the selection. If there are too many wrong answers, it decreases the overall efficiency of the Question Analysis process. DeepQA generates the hypotheses using components for search and candidate generation.

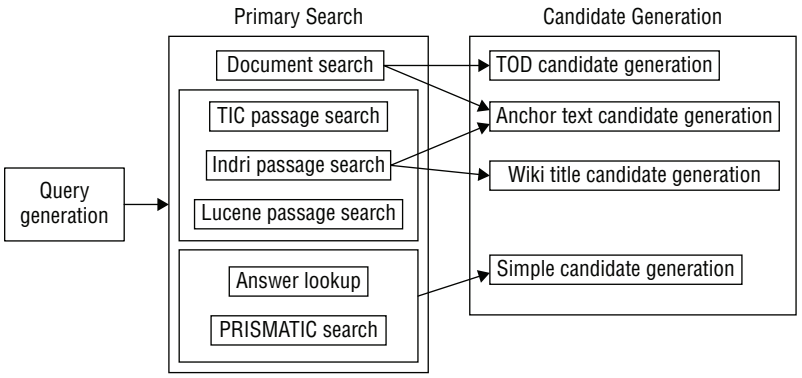

Figure 9-3: Hypothesis Generation in Watson's DeepQA Architecture

Following are two components:

- **Search**—Content that is relevant to the question is retrieved from Watson's corpus using search tools such as Apache Lucene. IBM developed highly effective and time-efficient search strategies that leverage the relationship between the content in documents and the titles of those documents. IBM enhanced the native capabilities in the search engines to improve results

by extracting syntactic and semantic relations for questions and resource sources.

- **Candidate generation**—Hundreds of potential candidate answers to a question are identified from the search results. Watson uses the knowledge in human-generated text and metadata, as well as syntactic and lexical cues in search results.

Referring to Figure 9-3, you can see that the DeepQA architecture relies on multiple search engines including Indri, PRISMATIC, and Lucene to index and search unstructured text and documents and then generate candidate answers. Each approach has certain benefits, and IBM has optimized results by combining the different approaches. For example, one of the key benefits of Apache Lucene for searching in Watson is flexibility in its architecture that enables its API to be independent of the file format. Apache Lucene is an open source text indexer and search engine written in Java. Text from the different types of sources in Watson's corpus (PDF, HTML, Microsoft Word, and so on) can all be indexed. This approach works for the corpus developed for *Jeopardy!*, as well as for commercial applications.

Scoring and Confidence Estimation

Scoring and confidence estimation is the final stage in the pipeline. The way Watson uses confidence estimation is a critical element in achieving a high level of accuracy. No single component of the system needs to be perfect. All candidate answers are ranked based on evidence scores, and these scores are used to select the answer with the greatest likelihood of being correct. The various passage-scoring methods used by Watson are combined to improve accuracy. Scores are assigned by matching question terms to passage terms. The net result of this approach to evaluating and ranking the answers ensures that the best answer comes out on top.

There are two methods used for domain relation extraction and scoring in DeepQA: manual pattern specification and statistical methods for pattern specification. The manual approach has a high accuracy rate but takes longer because of the need to find humans with the domain knowledge and statistical experience to create rules for the new relations. Watson looks for the candidate answer by filtering out the noise. There are many different models, including Hidden Markov models, which are used to filter out noise that does not fit the pattern.

There are many scoring algorithms. The four passage scoring (deep evidence scoring) algorithms used in this process are described next:

- **Passage Term Match**—This algorithm assigns a score by matching question terms to passage terms, regardless of grammatical relationship or word order.

- **Skip-Bigram**—This algorithm assigns a score based on relationships observed between specific terms in the question and terms in the evidence passage.
- **Textual Alignment**—This algorithm assigns a score by looking at the relationship between the words and word order of the passage and the question. The focus is replaced by the candidate answer.
- **Logical Form Answer Candidate Scorer (LFACS)**—This algorithm assigns a score based on the relationship between the structure of the question and the structure of the passage. The focus is aligned to the candidate answer.

The candidate answers are scored in parallel across a large cluster of machines, which speeds up the process significantly. This is one of many places within the DeepQA architecture where parallelism comes into play. This ensures that Watson maintains both speed and accuracy. Implementing these scoring strategies together yields better results than if each one was used individually. For example, LFACS is less effective than other algorithms when used individually. However, when used in combination with the other scoring methods, it helps improve overall effectiveness. Ultimately, the way Watson combines the multiple scoring algorithms is by using machine learning and training on questions with known correct answers.

Summary

IBM's Watson is a cognitive system designed to help expand the boundaries of human cognition. It represents a new era in computing technology by enabling people to begin to interact more naturally with computers. In this new era, humans can leverage and share knowledge in new ways. Watson makes it possible for humans to ask questions in natural language and get answers that enable them to gain new insights from extremely large volumes of information. The research for Watson was based on IBM's extensive experience in NLP, AI, information retrieval, big data, machine learning, and computational linguistics.

A cognitive computing system is not a simple automated processing system. It is intended to create new levels of collaboration between man and machine. Although humans have been codifying information for a long time, there are limitations on the insights and analysis that humans can glean from that information using traditional forms of computing. With a cognitive system like IBM's Watson, the machine can find patterns or outliers in large volumes of unstructured and structured information at fast speeds. A cognitive system gets smarter as each successive interaction improves accuracy and predictive

power. The relationship between people and machines is symbiotic in a cognitive system. Good results from a cognitive system require that humans do some mapping and training using machine learning techniques. Humans train Watson by building a corpus of knowledge that may be broad-based or fine-tuned to a specific area such as medicine or finance. The corpus includes information that is codified in books, encyclopedias, research studies, and ontologies. Watson can then search through vast quantities of information and analyze that data in order to provide accurate answers with confidence levels. IBM is applying Watson technologies to multiple industry solutions in fields such as healthcare, finance, and retail.