*Cognitive Computing and Big Data Analytics*
By Judith Hurwitz, Marcia Kaufman and Adrian Bowles

CHAPTER

10

# The Process of Building a Cognitive Application

Organizations in many different industries are in the early stages of developing cognitive applications. From healthcare to manufacturing to governments, decision makers need to quickly make sense of large volumes and varieties of data. Problem solving often requires the aggregation of a multitude of disconnected data sources including a combination of internal and external data. In addition, it is increasingly likely that the data required to answer problems or deliver new insights is unstructured—such as text, videos, images, sound, or sensor data. Valuable insights may remain hidden because the volume, variety, and velocity (speed) of the data are so hard to manage. Organizations are now beginning to recognize the potential benefits of using cognitive applications to find the patterns in data that can help to improve outcomes.

Chapters 11–13 provide examples of emerging cognitive computing applications across multiple industries. Although the domains and applications described in these chapters differ, certain common attributes of each situation make them a good fit for cognitive applications. Organizations that are implementing cognitive applications typically face similar challenges regarding data and the decision making process such as:

- Large volumes of unstructured data that must be analyzed to make good decisions.
- Decisions must be based on constantly changing data, new sources, and forms of data.

- A significant amount of knowledge about the domain is transferred from senior experts to trainees through a mentoring and training process.
- Decision making requires the analysis of a variety of options and solutions to a problem. Individuals often have to quickly weigh the relative risks and benefits of each alternative and may have to decide based on confidence rather than certainty.

This chapter examines the seven key steps involved in designing a typical cognitive application:

1. Defining the objective
2. Defining the domain
3. Understanding the intended users and their attributes
4. Defining questions and exploring insights
5. Acquiring the relevant data sources
6. Creating and refining the corpora
7. Training and testing

## The Emerging Cognitive Platform

The majority of early cognitive applications have been built from scratch by vendors in collaboration with their customers. The vendors and customers were experimenting and learning together. As the number of cognitive applications under development and deployment grows, vendors are using their experience to codify packaged services, APIs and delivery models that can help customers build cognitive applications more independently and quickly. Most new cognitive applications are developed on a cloud-based cognitive engine that provides the ability to scale processing, storage, and memory. In addition, customers will access a set of well-defined foundational services to speed development of cognitive applications. These foundational services may include a corpus service, analytics service, a data engine such as a graph database, training services, presentation and visualization services, and others. The expectation is that moving forward, cognitive applications will be built on an engine and well-defined APIs that provide some or all these foundational services.

In many ways, vendors are collaborating with partners on these early cognitive applications in a role similar to a systems integrator. Vendors are responsible for the development of the cognitive engine, but much of the development of associated tools and services are created jointly with partners based on their requirements. The partners begin by focusing on defining the domain for their cognitive application, collecting and curating the data sources, and understanding the types of questions and information that their users will be interested in. Typically, the development of the model, including the training and testing of the system, is completed in collaboration with the vendors providing the cognitive platform.

Each phase of developing a cognitive application can be time-intensive and requires the input of domain experts and end users. Many initiatives have required a significant amount of manual intervention in areas such as building and refining the corpus and training and testing the system. If cognitive applications are going to become more accepted and deliver value across many industries, vendors need to provide packages and tools that enable customers to get new applications up and running quickly. One of the most time-consuming aspects of building a cognitive application is selecting, accessing, acquiring, and preparing data for the corpus. Therefore, vendors are beginning to offer corpus services that include industry-specific, pre-ingested, and curated data. For example, in the healthcare industry, these sources might include a healthcare-specific semantic taxonomy and ontology of disease codes and symptoms. Training is critical to the success of the applications and can also be time-intensive. Vendors could provide a pretrained data set for the application in a particular domain or problem area. There will also be extensive use of APIs that abstract some of the challenging aspects of developing and maintaining the application. For example, APIs can simplify the process of importing data for visualization rendering or extracting relationships from data.

## Defining the Objective

Creating a cognitive application has much in common with developing any other enterprise application. You need to understand what the objectives are for your application and how you will achieve those objectives. Therefore, the first step in developing a cognitive application is to understand the types of problems your cognitive application is going to solve. Your objective needs to consider the types of users you will be appealing to and if there will be multiple constituencies in your user base. What issues will your users be interested in, and what do they need to know? One of the unique differences between a cognitive application and traditional applications is that users should expect more than answers to queries. A cognitive application should provide answers to questions but also go deeper and explore context related to how and why something happened.

Building traditional applications often begins with business process. In contrast, in a cognitive application you need to develop an objective based on knowledge and data. Therefore, in the design you need to set some parameters around the type of knowledge that is pivotal to your corpora. In other words, your objective should probably focus on a specific segment of an industry rather than attempt to solve all problems for a particular industry. Several examples of objectives for cognitive healthcare applications follow:

- Provide personalized information and social support to help individuals optimize their health and wellness.
- Help health consumers take a more active role in managing their own health and the health of people they help care for.
- Help determine if the treatment plan selected for the patient is the best and most cost-effective.
- Provide additional knowledge to medical students to support what they learn on their subspecialty rotations.

Cognitive applications are also a good solution in situations in which you would like to provide assistance to a customer service representative or salesperson. Consider a retail organization with a large and diverse group of sellers. This company has a small number of sellers with many years of experience and deep knowledge of the products the company sells. If a customer has specific requests or needs help comparing alternatives, these knowledgeable sellers can answer all their questions and make the sale. However, the company also has high turnover, and many sellers lack the knowledge to provide the right level of support to customers. The company decides to introduce a cognitive application designed to make all the sellers as smart as the most knowledgeable and experienced seller in the company.

## Defining the Domain

The next step is to specify the domain or subject area for your cognitive application. Defining your domain is a prerequisite to identify and assess which data sources you will need for your applications. In addition, your domain definition will be useful in determining the subject matter experts that will be helpful in training the system. Table 10-1 provides a few examples of cognitive application domains and a sample of the data sources and subject matter experts that can create the knowledge base for that domain. As described in the previous section, your stated objective is likely to help narrow the domain focus. For example, a cognitive application designed to train medical students would require medicine as the domain, whereas a cognitive application designed to help clinicians select the right treatment plan for breast cancer patients would require breast oncology as the domain. The medical domain will require comprehensive and broad-based medical taxonomies, ontologies, and catalogues, whereas the breast oncology domain will require segments of the medical ontology as well as additional data specific to the field.

**Table 10-1:** Examples of Cognitive Application Domains

| DOMAIN | SELECTED DATA REQUIREMENTS | SUBJECT MATTER EXPERTS |
|---|---|---|
| Medicine | International Classification of Diseases (ICD) codes, Electronic Medical Records (EMR), and research journals | Senior physicians in medicine and key specialties |
| Airplane Manufacturing and Maintenance | Complete parts list, maintenance records per airplane, and spare parts inventory | Mechanics who know how to anticipate failure and create effective repairs and experienced pilots |
| Retail | Customer and product data | Experienced sales associates |

Although the domain helps to define the data sources you might need, you may also include data sources that are not typically associated with solving problems in that domain. The inclusion of non-typical data sources is needed because cognitive systems support problem solving in a different way from traditional systems. A cognitive application is good at helping users to assimilate knowledge quickly and efficiently. Although some of this knowledge might be found in specific data sources, it may also incorporate information that is typically learned by experience. One of the advantages of a cognitive application is that it can provide every user with proven business practices and industry-specific knowledge that is well known to your most experienced domain experts. A cognitive system's greatest value comes from its ability to combine information from industry data sources with testing and refinement based on interactions with highly experienced experts. For example, when faced with an unusual problem, an airplane mechanic with 30 years of experience might remember similar situations that occurred in the past and recommend something like, "the problem is likely to be A or B and we should take these five steps to get the best result."

## Understanding the Intended Users and Defining their Attributes

You need to understand the types of users who will be accessing your cognitive application. Expectations for user and system interactions will have an impact on the development of the corpus, the design of the user interface, and how the system is trained. The level of accuracy required in a cognitive application will depend on the intended use case. For example, a scientist requires a much more precise level of accuracy than a customer service representative answering questions about replacement parts. However, it is unnecessary and unwise to attempt to anticipate all the questions your users will ask and all the different ways your cognitive application will be used. A cognitive application assumes that the data will grow and change as new data sources are discovered and added. In addition, the machine learning algorithms will refine the way questions are analyzed and answered. You need to build in flexibility so that your application can change as user requirements change. The learning process for a cognitive system is continuous, and as a result, your application will get smarter and deliver greater value to end users the more it is used.

The following best practices can help to ensure that your cognitive application has the flexibility it needs to provide the right level of support for its users:

- **Understand your users' level of understanding of the domain.** Is your cognitive application intended for consumers or domain experts? Will your users understand the meaning of industry-specific terms? Will your cognitive application be used to help train users in a particular domain?
- **Plan for variations in types of questions and analysis required.** Will your application have users with varied backgrounds and levels of expertise? For example, if you are planning for consumers and domain experts, they are likely to ask questions using different words and language styles. Although these users may be looking for insights into a similar topic, they may have widely differing expectations for the level of insight required. The consumer may look for a definition, whereas the domain expert wants to compare alternative solutions to a complex problem.
- **Keep the scope of your application broad enough to support different types of users.** If you are too specific or narrow in your definition of the domain, there may be subject areas that are not covered adequately in the corpus. It is better to err on the side of a little more coverage of your domain than less, as the learning process will successively refine the corpus toward the "right size" with increased usage.

## Defining Questions and Exploring Insights

As discussed in Chapter 1, "The Foundation of Cognitive Computing," a cognitive system delivers insight relative to a domain, a topic, a person, or an issue, based on training and observations from all varieties, volumes, and velocities of data. A cognitive system creates models to represent the domain and generates and scores hypotheses to answer questions or provide insight. To ensure your cognitive application delivers the insights your users are looking for, you need to begin by mapping out the types of questions they may ask. Users of a well-defined and trained cognitive application can benefit in many ways. One significant benefit is the capability to receive alternative answers to questions along with associated confidence levels. These benefits can be achieved only if the right set of data for the domain and the system is ingested into the corpus and then is properly trained and tested. However, before you even begin to train the system, you need to consider the types of questions your users will ask and the types of insight that your users will be looking for.

Many of the early cognitive applications are of two main types: customer engagement, or discovery and exploration. Customer or user engagement applications typically leverage advanced Question-Answer Systems designed to answer questions as part of an ongoing dialogue with the user. Answers to questions may be provided as a set of alternatives with associated confidence levels. Discovery and exploration applications begin with data analysis rather than by asking questions. You may not know what to expect or exactly what questions to ask. Discovery applications are used in situations such as genomic exploration, security analysis, or threat prevention. Typically, in these situations, your cognitive application will begin by looking for patterns and anomalies in the data.

Because the Question-Analysis approach to cognitive applications requires a more rigid structure to understand potential user questions, this process is described next. All questions need to be suitable for evidenced-based analysis; however, the questions do not all need to be initiated by the user. Actually, one of the defining features of a cognitive application is that users can engage in a dialogue with the system. In anticipatory systems, the application is designed to analyze the data and make suggestions or recommendations for the user without the user needing to ask a specific question. As a result, the user can move through paths of analysis not previously anticipated and develop new insights based on the user/application interaction. The cognitive system can make associations between questions, answers, and content to help the user understand the subject matter at a deeper level. The questions users will ask can be placed in two general categories:

- **Question-Answer pairs**—The answers to these questions can be found in a data source. There may be conflicting answers within the data sources, and the cognitive system will analyze the alternatives to provide multiple responses with associated confidence levels.
- **Anticipatory analytics**—The user engages in a dialogue with the cognitive application. The user may ask some questions but not all the questions. The cognitive application will use predictive models to anticipate the user's next question or series of questions.

## Typical Question-Answer Pairs

Developers of question-answer cognitive applications have found that they need to begin with approximately 1,000–2,000 pairs of questions and answers. You have already defined the users of your application, and you need to keep them in mind when creating the question-answer pairs. How will your representative group of users ask questions? Consider not only the content of the question, but also how it will be asked. The questions need to be in the voice of the end user. What style of language will they use? What technical terms are they likely to know? There are often many ways to ask the same question, and you need to consider the alternative styles of questioning when developing these initial questions. Although the answers need to use terms and a language style that will be understood by users, the content of the answer needs to be vetted by subject matter experts.

Table 10-2 provides an example of two questions that might be asked of a medical cognitive application, related to the use of morcellators: a health consumer asks one question and a gynecologist asks the other. The health consumer is looking for a definition, whereas the health professional is looking for more details on risks and benefits of a specific procedure. In a cognitive application, users of both types could engage in a dialogue that would provide more granular information on the topic.

**Table 10-2:** Question-Answer Pairs for Different Types of Users

| QUESTION | ANSWER |
|---|---|
| Heath consumer: What is a morcellator? | A morcellator is a device with a spinning blade that is used to shred a fibroid through an incision on a woman's abdomen. The force and speed of the device may cause cellular particles from the fibroid to become dispersed in the abdomen. |
| Gynecologist: What are the risks and benefits of using a morcellator for surgical treatment of fibroids? | Risks include potential spread of an occult uterine sarcoma. Benefits include smaller incisions for the patient, less bleeding, and quicker healing and recovery. |

You should define a sample set of questions prior to selecting the data sources needed to build the corpus. By choosing your information sources based on what is needed to answer a representative set of questions, your system can learn how to answer similar questions in the same domain. If you build the corpus first, you may make the mistake of tailoring your questions for training and testing to the information you already have at hand. When your cognitive application becomes operational, your users may have questions that cannot be answered by the system. It is expected that the corpus will need to be continuously updated during training and operation; however, you want to start out by including as many data sources as required to provide the right level of insight within your chosen domain.

## Anticipatory Analytics

What if the user is not in a position to ask a specific question of the cognitive application? Anticipatory analytics can be used when there are many unknown factors making it difficult for a user to know what questions to ask. For example, in military or security analytics, you may not know when or where a future event will occur or even what event will occur. You need to observe the data and look for patterns without knowing what you are looking for. The data you need to observe and analyze may be unclean or subject to inconsistent definitions and inconsistencies in metrics or measurements around time and place. However, when this data is used for a cognitive security application, unclean data may provide valuable clues to anticipate events or actions. The anomalies or outliers in the data are used to build the models and anticipate changes that can identify security threats or military events in time to take corrective action.

Anticipatory analytics is also used in cognitive applications that are designed to understand an individual's personal needs and help them to make good decisions. Because a user does not need to ask a question to be provided with a recommended action, the creators of the application need to focus on the different personal situations that might be best suited for assistance by a cognitive system. For example, a cognitive assistant could monitor a user's schedule and alert the user if there is a delay in a scheduled air flight or train. By monitoring personal medical devices and applications, a cognitive assistant could alert users they may be getting sick or help them keep on track with dietary goals. Users are increasingly sharing a lot of personal information on a variety of applications and devices—ranging from health monitoring devices to e-mail, travel, and calendar applications. A cognitive application can be trained to integrate this information to learn a lot about you. In addition, a cognitive application can be designed to be aware of what is happening in the world around you through geospatial, travel, health, and other applications. Therefore, a cognitive application that understands your location, your health and medical status, and the context of your questions can make personalized recommendations. An anticipatory cognitive application leverages data to make personal tasks easier and provide information you need before you ask for it.

## COGNITIVE COMMERCE

**Cognitive commerce refers to a cognitive application designed to anticipate user needs from a retail or commerce perspective. Organizations with mobile or Internet-based commerce sites are continuously trying to optimize their sites to increase sales. By making it easier for consumers to find what they want faster, these companies can reach their sales goals faster. For example, a company that provides streaming entertainment content could create a cognitive application to make it easier for customers to find the movie they want to watch and make it easier to watch on their mobile device.**

**Cognitive capabilities are built in to an existing commercial app or other environment. The user has previously provided permission to the commercial**

*Continues*

*(continued)*

**application to capture personal information (that is, health data, travel itinerary, and exercise tracking). As a result, the application can make suggestions or provide information to the user without the user needing to ask specific questions.**

**Builders of a cognitive application with commerce capabilities need to plan for the types of questions users will ask as well as the types of capabilities that will have a positive impact on sales. For example, you may expect that users will ask a question about ordering a specific item such as, "Do you have XBrand jeans available in dark wash size 29?" However, you may also want to plan for questions that are more open ended such as, "I saw the perfect silk dress on 'X' character in 'Y' on 'ABC' show. Can you find me something similar in size 4?" You may also want to submit a photographic image of a dress and ask the system to locate the item in a different color or size. A cognitive commerce application could accept complex user queries in natural language and make it much easier and faster for consumers to find the right item to purchase. In addition, by understanding your personal information in context, a cognitive commerce application can anticipate what you might like to buy next before you do.**

## Acquiring the Relevant Data Sources

When developing a corpus you should determine the most relevant data sources. This is challenging because you cannot know with certainty what type of insights users might require as their needs change over time. However, taking the time to evaluate data sources you currently own and those you may want to acquire also offers great opportunity. You may discover that you have internal data resources that can provide new insight when leveraged by a cognitive system. Additionally, you may want to include social media data or other external sources. Cognitive systems provide an opportunity to leverage data sources in new ways. To start building the corpus, you need to understand your requirements for a variety of internal and external data sources. As you move through the testing process and your application becomes operational, you need to be prepared to add new sources as they become available and the scope of the application expands.

### *The Importance of Leveraging Structured Data Sources*

Much of the focus around cognitive computing has been data from unstructured data sources. However, cognitive solutions must gain insights based on the current state of customers or other constituents. Therefore, you need to know what internal data sources are going to be meaningful. For example, if the application is related to travel, the company needs internal data to relate to the details about customers or travel locations. A retail application needs data sources related to merchandise that has been ordered, what products have been sold, and who the customers are. A hospital-based healthcare application needs data on patient status, medical history, and hospital admissions. A manufacturing application may need data that reports on sensor activity from the production floor. These data sources will most likely be stored as structured data in relational databases including customer data from a Customer Relationship Management system or patient data from an Electronic Medical Record for a healthcare application. Additionally, there could be streaming data sources that come from sensor networks.

### Analyzing Dark Data

*Dark data* refers to data that has been stored over many years and sometimes decades. Much of this data has been stored but not previously analyzed. For example, dark data could be data about performance of a company's stock over a decade or data stored at the time of a security breach. With the cognitive system, the dark data can become the benchmark to analyze how things have changed over time. This data may provide new insights by using machine learning to look for patterns in data collected over many years. Given the advent of new analytics technologies, this dark data may now be an important internal data source depending on the domain.

### Leveraging External Data

What external data sources will support users? External data sources may include everything from industry-specific technical journals that are focused on new research findings to industry taxonomies and ontologies. In medical research there are results from clinical trials that might provide insights into drug interactions. Most industries have a wealth of third-party databases with both structured and unstructured data. Increasingly, there are stores of videos, images, and sounds that are of particular interest to either a specific industry or a technical discipline.

Many industries have codified ontologies and taxonomies that are managed and updated by industry consortiums. These sources are critical in creating your corpus. However, you may find that you need to capture only a subset of the available data. These data sources often include the hierarchical classification of entities or concepts within a domain, which are important for determining context and meaning. Table 10-3 provides you with a sample of the types of ontologies and taxonomies available for certain specific industries.

You need to use caution when using these external data sources. For example, what is the origin of the data source? Who owns that data source and how and when was it created? More important, who is responsible for updating the data source on an ongoing basis? Equally important is the security and governance of the data sources. There are data sources that include private information that can be used under strict governance guidelines. If that data is misused, it can cause significant problems for an organization.

**Table 10-3:** Industry-Specific Taxonomies and Ontologies

| INDUSTRY | TAXONOMY/ ONTOLOGY | PUBLISHER | DESCRIPTION |
|---|---|---|---|
| Healthcare | International Classification of Diseases (ICD) | World Health Organization | International codes for diseases, disease symptoms, and medical findings about diseases |
| Healthcare | Semantic taxonomy for the healthcare ecosystem | Developed by companies such as Healthline Corp. | Classifies healthcare information on the web and maps the relationship between consumer and clinical terminology |
| Construction | International Building Code (IBC) | International Conference of Building Officials | Standards and compliance regulations for international building codes |
| Finance | U.S. GAAP Financial Taxonomy | Financial Accounting Standards Board (FASB) | U.S.-based standards for financial accounting and reporting |
| Information Technology | NIST Cloud Computing Taxonomy | National Institute of Standards and Technology (NIST) | Companion to the NIST Cloud Computing Reference Architecture; goal to help communicate the offerings and components of cloud architecture |

## Creating and Refining the Corpora

Building a cognitive application requires extensive collaboration between the technology team and business experts. The initial steps in the development process include defining the objective and user expectations for the application. This stage requires substantial industry or domain expertise. The next series of steps in the application development process relies more heavily on the technology team. The actual creation of the corpus, model development, and training and testing of the system requires skills in areas such as software development, machine learning, and data mining.

The creation of the corpus is not a one-time process. There is an initial effort to build a quality corpus (or corpora) that includes the selected data sources. However, there needs to be continuous re-evaluation of the data sources to determine if new sources need to be added or if enhancements to existing sources are required to improve outcomes from the cognitive application. You need to understand the life cycle for each of the data sources because many of these sources need to be updated at regular intervals. Therefore, you need to set a process in place to ensure that updates to data sources are made on a timely basis.

Although a cognitive application leverages data from the corpora as its primary base of knowledge, not all the data sources used by the system need to be ingested into the corpus. Much of the data may be called as a cloud-based service and used by the application without being included in the corpus. A cognitive application may need to interact with a variety of data management systems including Hadoop, column store, graph, and other environments.

The process of creating the corpus includes preparing the data, ingesting the data, refining the data, and governing that data throughout its life cycle. These steps are described in the following sections.

## Preparing the Data

All data ingested into the corpus must first be validated to ensure that it is readable, searchable, and comprehensible. As detailed in the previous section, structured, semistructured, and unstructured data are likely to be combined in various corpora included in a cognitive system. All data sources need to be evaluated to see if any transformations or enhancements are required prior to ingestion into a corpus. Are your text-based resources such as journal articles, textbooks, and research documents annotated with headings that provide queues to the cognitive system? Tagging should help the system identify and classify the content in specific articles. In addition, tagging can ensure that the cognitive system can quickly make appropriate associations between different data elements.

The requirements to transform the structure of the data can vary based on the cognitive platform you use. The corpora of some early cognitive systems were ingested with primarily unstructured text-based content. As a result, complex structured data sources needed to be transformed into unstructured content prior to ingestion into the corpus. Initially, this transformation was time-consuming. However, services have been developed to help speed up the process of transforming data structures. Vendors are continuing to improve data preparation services for cognitive systems, making it possible for structured data to be automatically transformed within the system. These transformation and other data preparation services can have a positive impact on adoption rates for cognitive applications. Data from structured data sources such as Customer Relationship Management Systems or other database applications needs to be easily and quickly ingested into the system if business users are going to begin using the applications at a greater rate. It is not necessary that the complete database be ingested as is. Actually, it is quite common that only a segment of the existing data source is required to meet the requirements for the domain.

## Ingesting the Data

Managing the data ingestion process efficiently is critical to the success of a cognitive application. Data ingestion is not something that happens just once during the development of the system. Existing data sources are subject to continuous updates and refinements to ensure they are accurate and up to date. The results of the training and testing of the models may indicate weak spots or limitations in the corpora that require the addition and revision of sources. In addition, changing user expectations are likely to result in new additions to the corpora. Delays in making the required updates to the corpora will decrease the effectiveness and accuracy of the system. Therefore, to maintain the viability of the cognitive system, data sources may need to be ingested in near real time. Typically, you will have access to a set of services that are designed to make the ingestion process fast, robust, and flexible. Although there may be some coding required, the ingestion services will include connectors and tools to make the process as seamless as possible.

As in traditional data management efforts, you need to have controls and supports in place to maintain governance and anticipate and correct for errors. For example, you must incorporate real-time traceability into the data ingestion process. If errors result in an unexpected halt in the ingestion process, you need to trace back to understand why the problem occurred and where you were in the ingestion process when it stopped. This is called *checkpointing*, and you can then use this information to restart the ingestion process in the right place. In addition, you may need to monitor the ingestion process to ensure that any records that are deleted or scrubbed to meet security requirements have been handled properly.

## Refining and Expanding the Corpora

As mentioned in the previous section, a corpus needs to be continually refined to ensure that the cognitive application delivers accurate information and provides the right level of insight. Although you have completed extensive preparation for ingesting content needed to provide a good knowledge base for your cognitive application, it is hard to anticipate all data requirements at the outset.

Early in the training process, you may find that the accuracy of the answer to a certain question is below your accepted threshold. By increasing the coverage (adding more data) for certain topic areas in your domain, you should improve accuracy. Plan for multiple iterations of this process of training, observing results, and then adding to the corpus. You need to establish an ongoing process of updating data requirements and adding to the corpus as you proceed through the testing process and after your application becomes operational. You can use expansion algorithms to determine which additional information would do the best job of filling in gaps and adding nuance to the information sources in the corpus. There will be situations in which you need to enrich data by providing lookups to additional sources that might have detailed information about customers or definitions of technical data.

## Governance of Data

The corpora in your cognitive application will include a wide range of data sources. There may be personal data that is subject to the same data privacy rules that apply to data used in other systems in your organization. Therefore, you will need to comply with the same data privacy and security requirements of any system. There will be data that will be ingested into the corpus that might have restrictions on use based on governance requirements. In some situations there might be copyrighted images or content that is part of your corpus. Therefore, you want to make sure that you have a license for use of that content. In healthcare there are patient privacy rules that require that personal information be anonymized. In a retail system it will be important not to expose customers' credit card data. If a corpus includes social media data, you must be sure that you are not violating the privacy of users of those sites. For example, users might decide that they no longer want to allow access to location data. In some countries there are restrictions on where customer data can be stored. A cognitive system may require the highest level of governance and security because over time it will include sensitive data about competitive best practices. Therefore, in designing and operating a cognitive system, governance and security cannot be an afterthought.

## Training and Testing

It is through an iterative process of model development, analysis, training, and testing that the cognitive system begins to learn. Deploying a scalable training and testing strategy can ensure your application works as intended when it becomes operational. You need to measure responses to determine what is the minimal level of accuracy that is acceptable. After this is established through the testing process, you can begin to establish the *ground truth*—a set of data that is the gold standard for accuracy of a model. It may require you to try additional data sets so that the information used for testing is objective. Initially, you create a ground truth that establishes what the system knows and understands. In Question-Answer based cognitive applications, you have a set of question-answer pairs that establish your ground truth. The questions represent the types of questions your users will ask. The answers to those questions are accurate, having been approved by domain experts. These question-answer pairs are developed in clusters around a topic to help with the machine learning process. Algorithms help the system to understand context by looking for associations and patterns in the clusters of question-answer pairs. Your training and testing strategy needs to compare new analysis against the ground truth and add to the base level of truth when needed to improve the accuracy of the system.

This is often an iterative process; each time the data is trained, the accuracy of the application improves.

Cognitive systems are designed to learn from failure and improve through feedback. Your cognitive application may assign a high confidence level to answers that are obviously wrong. As part of the training process, you need to analyze why the system got the answer wrong. Although training the system should be an automated process, there are some aspects that involve manual intervention, particularly by subject matter experts. Figure 10-1 illustrates the steps that help to analyze the reason for the error and what corrective action to take to improve accuracy in the future. These errors are measured against key measures for monitoring cognitive system performance including recall, precision, and accuracy.

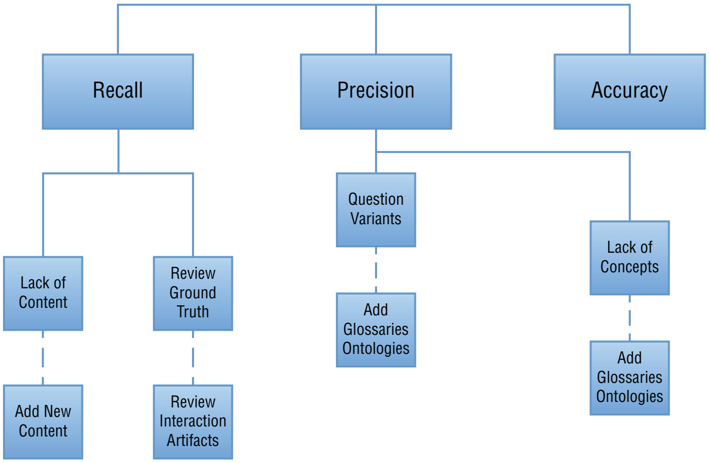

**Figure 10-1:** Improving accuracy of the models

The section "Creating and Refining the Corpora" earlier in this chapter detailed the importance of adding and updating data to ensure the corpus can support the cognitive application. However, lack of data is not the only reason why an application may provide incorrect answers. Subject matter experts may need to review the ground truth and make adjustments to the answers provided to the system. Other errors may occur because the models cannot capture some of the relationships and nuances between similar data sources. One approach to improve this is by adding glossaries and ontologies that provide the system with cues to learn more about key concepts.

Training and testing data can be one of the most time-consuming parts of the process of creating a cognitive system. The smaller the domain, the easier it will be to both create the corpus and find training data to ensure that the information can answer questions and learn over time. In this situation, you can select a sample data set that is representative of the type of questions and type of problems you are addressing. If the domain is larger and more complex, it requires a larger set of sample data. In many situations, you can select sample data that is directly applicable to the problem. For example, the data regarding consumer questions about treatment of diabetes are well understood. However, you may have a situation in which there isn't a lot of certainty regarding outcomes. For example, if you want to understand data from traffic management in a large metropolitan area, you might need a vast amount of sensor data. You may not select the right set of data that is representative of the patterns you want to identify. As you can see, training and then testing results can be complicated by scope and scale issues.

The most important part of the training process is to have enough data so that you are in a position to test your hypothesis. Often the first pass at training provides mixed results. This means that you either might need to refine your hypothesis or provide more data. This process is not unlike learning any new discipline where you start with your assumptions based on incomplete knowledge. As you learn more, you can determine that you need more data from more sources. As you gain more insights from the data, your assumptions will change. At this point you are ready to test your understanding of the domain to see if you have the right amount of knowledge or if you are still required to collect more data and learn more. This is precisely what happens in an automated fashion when you design a cognitive system.

## Summary

Implementing a cognitive solution is a multistep process that begins with understanding the goals and objectives of the project. These steps begin by establishing your objectives: the domain and key user attributes. You also have to define the type of questions you expect users to ask and what insight they may be looking for. You are also required to determine and find the relevant data sources both from internal and external sources. After these stages are complete, you create and refine the corpora. The final stage is the training and testing process. But keep in mind that this is not a serial process. Building a cognitive system is iterative because data continues to change, and the nature and attributes of users changes. A well-designed cognitive system can become a new model for gaining significant insights into business knowledge.