Chapter 13

Understanding Text Analytics and Big Data

In This Chapter

- Exploring the different types of unstructured data
- Defining text analytics
- Unstructured analytics use cases
- Putting unstructured data together with structured data
- Text analytics tools for big data

Most data is unstructured. Unstructured data includes information stored internally, such as documents, e-mails, and customer correspondence, as well as external information sources that are important to your organization, such as tweets, blogs, YouTube videos, and satellite imagery. The amount and variety of this data are growing rapidly. Increasingly, companies want to take advantage of this wealth of data to understand the implications for their business today and in the future.

While image and audio analysis are still in the early adopter stage, text analytics is evolving into a mainstream technology. Here's an example of how one company was able to leverage its text data to support business decision making. A large automobile manufacturer needed to improve quality problems with its cars. It discovered that by analyzing the text from its repair partners, it could identify quality problems with its cars as they enter the marketplace. The company views this analysis as an early warning system. The earlier it can identify problems, the more changes it can make on the factory floor and the fewer customers will be dissatisfied. Prior to using text analytics, the company mined information from its line of business systems, including part numbers and defect codes. This worked well enough for many years, but only for problems the company already knew existed. The traditional system could not reveal hidden issues that were well known to the people who were interacting with customers.

Sounds exciting, right? In fact, text analytics is being used in a wide variety of big data use cases from social media analysis to warranty analysis to fraud analysis. In addition, businesses are increasingly beginning to analyze a merge view of structured and unstructured data together to get a full picture. In this chapter, we delve into this technology and provide an in-depth example of how it works. We also provide you with some other use cases of text analytics in action, including the capability to merge unstructured data with structured data. We end the chapter with the names of some vendors that are providing text analytics tools for big data.

Exploring Unstructured Data

What sets unstructured data apart from structured data is that its structure is unpredictable. As we mention in Chapter 2, some people believe that the term *unstructured data* is misleading because each text source may contain its own specific structure or formatting based on the software that created it. In fact, it is the content of the document that is really unstructured.

Just think about the kinds of text that are out there and the structure that might be associated with each:

- **Documents:**

  In return for a loan that I have received, I promise to pay $2,000 (this amount is called *principal*), plus interest, to the order of the lender. The lender is First Bank. I will make all payments under this note in the form of cash, check, or money order. I understand that the lender may transfer this note. The lender or anyone who takes this note by transfer and who is entitled . . .

- **E-mails:**

  Hi Sam. How are you coming with the chapter on big data for the *For Dummies* book? It is due on Friday.

  Joanne

- **Log files:**

  ```
  222.222.222.222- - [08/Oct/2012:11:11:54 -0400] “GET / HTTP/1.1” 200
  10801 “http://www.google.com/search?q=log+analyzer&ie=…. . .
  ```

- **Tweets:**

  #Big data is the future of data!

- **Facebook posts:**

  LOL. What are you doing later? BFF

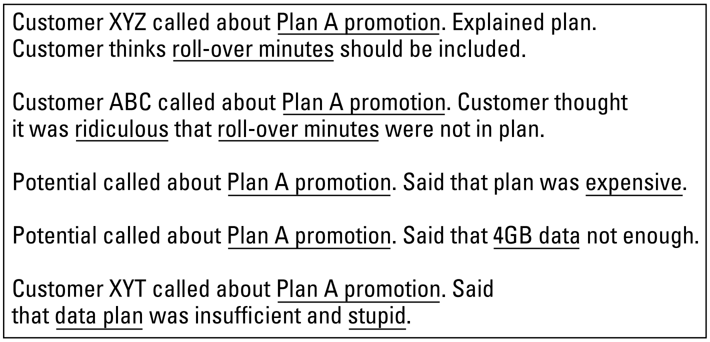
Clearly, some of these examples have more structure than others. For instance, a bank loan note has some structure in terms of sentences and the template it might follow. An e-mail might have little structure. A tweet or a Facebook message might have strange abbreviations or characters. A log file might have its own structure.

So, the question is, how do you analyze this disparate kind of unstructured text data?

Understanding Text Analytics

Numerous methods exist for analyzing unstructured data. Historically, these techniques came out of technical areas such as Natural Language Processing (NLP), knowledge discovery, data mining, information retrieval, and statistics. *Text analytics* is the process of analyzing unstructured text, extracting relevant information, and transforming it into structured information that can then be leveraged in various ways. The analysis and extraction processes take advantage of techniques that originated in computational linguistics, statistics, and other computer science disciplines.

Sometimes an example can help to explain a complex topic. Suppose that you work for the marketing department in a wireless phone company. You've just launched two new calling plans — Plan A and Plan B — and you are not getting the uptake you wanted on Plan A. The unstructured text from the call center notes might give you some insight as to why this happened. Figure 13-1 illustrates some of the call center notes.



Customer XYZ called about Plan A promotion. Explained plan.
Customer thinks roll-over minutes should be included.

Customer ABC called about Plan A promotion. Customer thought
it was ridiculous that roll-over minutes were not in plan.

Potential called about Plan A promotion. Said that plan was expensive.

Potential called about Plan A promotion. Said that 4GB data not enough.

Customer XYT called about Plan A promotion. Said
that data plan was insufficient and stupid.

Figure 13-1: Sample call center records.

The underlined words provide the information you might need to understand why Plan A isn't gaining rapid adoption. For example, the entity Plan A appears throughout the call center notes, indicating that the reports mention the plan. The terms *roll-over minutes, 4GB data, data plan,* and *expensive* are evidence that an issue exists with roll-over minutes, the data plan, and the price. Words like *ridiculous* and *stupid* provide insight into the caller sentiment, which in this case is negative.

The text analytics process uses various algorithms, such as understanding sentence structure, to analyze the unstructured text and then extract information, and transform that information into structured data. The structured data extracted from the unstructured text is illustrated in Table 13-1.

Table 13-1 Making Structured Data from Unstructured Text

Identifier	*Entity*	*Issue*	*Sentiment*
Cust XYZ	Plan A	Roll-over minutes	Neutral
Cust ABC	Plan A	Roll-over minutes	Negative
XXXX	Plan A	Expensive	Neutral
XXXX	Plan A	Data plan	Neutral
Cust XYT	Plan A	Data plan	Negative

You may look at this and say, “But I could have figured that out by looking at the call center records.” However, these are just a small subset of the information being recorded by thousands of call center agents. Each individual agent cannot possibly sense a broad trend regarding the problem with each plan being offered by the company. Agents do not have the time or requirement to share this information across all the other call center agents who may be getting similar numbers of calls about Plan A. However, after this information is aggregated and processed using text analytics algorithms, a trend may emerge from this unstructured data. That’s what makes text analytics so powerful.

The difference between text analytics and search

Notice that we are focusing on extracting text, not on keyword search. Search is about retrieving a document based on what end users already know they are looking for. Text analytics is about discovering information. While text analytics differs from search, it can augment search techniques. For example, text analytics combined with search can be used to provide better categorization or classification of documents and to produce abstracts or summaries of documents.

Table 13-2 illustrates four technologies: query, data mining, search, and text analytics. On the left side of the table are query and search, which are both about retrieval. For example, an end user could query a database to find out how many customers stopped using the company's services in the past month. The query would return a single number. Only by asking more and different queries will the end user get the information required to determine why customers are leaving. Likewise, keyword search allows the end user to find the documents that contain the names of a company's competitors. The search would return a group of documents. Only by reading the documents would the end user come up with any relevant answers to his or her questions.

Table 13-2 Query, Data Mining, Search, and Text Analytics

	Retrieval	*Insight*
Structured	Query: Returns data	Data mining: Insight from structured data
Unstructured	Search: Returns documents	Text analytics: Insight from text

The technologies on the left return pieces of information and require human interaction to synthesize and analyze that information. The technologies on the right — data mining (discussed in Chapter 12) and text analytics — deliver insight much more quickly. Hopefully, the value of text analytics to your organization is becoming clear.

Analysis and Extraction Techniques

Okay, now it's time to get a little bit more technical. In general, text analytics solutions use a combination of statistical and Natural Language Processing (NLP) techniques to extract information from unstructured data. NLP is a broad and complex field that has developed over the last 20 years. A primary goal of NLP is to derive meaning from text. Natural Language Processing generally makes use of linguistic concepts such as grammatical structures and parts of speech. Often, the idea behind this type of analytics is to determine who did what to whom, when, where, how, and why.

NLP performs analysis on text at different levels:

- **Lexical/morphological analysis** examines the characteristics of an individual word — including prefixes, suffixes, roots, and parts of speech (noun, verb, adjective, and so on) — information that will contribute to understanding what the word means in the context of the text provided. Lexical analysis depends on a dictionary, thesaurus, or any list of words that provides information about those words. In the case of a wireless communication company's sales promotion, a dictionary might provide the information that *promotion* is a noun that can mean an advancement in position, an advertising or publicity effort, or an effort to encourage someone's growth. Lexical analysis would also enable an application to recognize that *promotion, promotions,* and *promoting* are all versions of the same word and idea.
- **Syntactic analysis** uses grammatical structure to dissect the text and put individual words into context. Here you are widening your gaze from a single word to the phrase or the full sentence. This step might diagram the relationship between words (the grammar) or look for sequences of words that form correct sentences or for sequences of numbers that represent dates or monetary values. For example, the wireless communication company's call center records included this complaint: "The customer thought it was ridiculous that roll-over minutes were not in the plan." Syntactic analysis would tag the noun phrases in addition to providing the part-of-speech tags.
- **Semantic analysis** determines the possible meanings of a sentence. This can include examining word order and sentence structure and disambiguating words by relating the syntax found in the phrases, sentences, and paragraphs.
- **Discourse-level analysis** attempts to determine the meaning of text beyond the sentence level.

In practice, to extract information from various document sources, organizations sometimes need to develop rules. These rules can be simple:

> The name of a person must start with a capital letter.
>
> Every course on the college website must follow a three-digit course number and a semicolon.
>
> A logo must appear in a certain location on every page.

Of course, the rules can be much more complex. Organizations can generate rules manually, automatically, or by a combination of both approaches:

- In the manual approach, someone uses a proprietary language to build a series of rules for extraction. This person may also build dictionaries and/or synonym lists. While the manual approach can be time-consuming, it can provide very accurate results.
- Automated approaches may use machine learning or other statistical techniques. The software generates rules based on a set of training and text data. First, the system processes a set of similar documents (for example, newspaper articles) to develop — that is, learn — the rules. Then the user runs a set of test data to test the accuracy of the rules.

Understanding the extracted information

The techniques described earlier in the chapter are generally combined with other statistical or linguistic techniques to automate the tagging and markup of text documents to extract the following kinds of information:

- **Terms:** Another name for keywords.
- **Entities:** Often called *named entities,* these are specific examples of abstractions (tangible or intangible). Examples are names of persons, names of companies, geographical locations, contact information, dates, times, currencies, titles and positions, and so on. For example, text analytic software can extract the entity *Jane Doe* as a person referred to in the text being analyzed. The entity *March 3, 2007* can be extracted as a date, and so on. Many vendors provide entity extraction out of the box.
- **Facts:** Also called *relationships,* facts indicate the who/what/where relationships between two entities. John Smith is the *CEO of Company Y* and *Aspirin reduces fever* are examples of facts.
- **Events:** While some experts use the terms *fact, relationship,* and *event* interchangeably, others distinguish between events and facts, stating that events usually contain a time dimension and often cause facts to change. Examples include a change in management within a company or the status of a sales process.
- **Concepts:** These are sets of words and phrases that indicate a particular idea or topic with which the user is concerned. This can be done manually or by using statistical, rule-based, or hybrid approaches to categorization. For example, the concept *unhappy customer* may include the words *angry, disappointed,* and *confused* and the phrases *disconnect*

service, didn't call back, and *waste of money* — among many others. Thus the concept *unhappy customer* can be extracted even without the words *unhappy* or *customer* appearing in the text. Concepts can be defined by users to suit their particular needs.

- **Sentiments:** Sentiment analysis is used to identify viewpoints or emotions in the underlying text. Some techniques do this by classifying text as, for example, subjective (opinion) or objective (fact), using machine-learning or NLP techniques. Sentiment analysis has become very popular in "voice of the customer" kinds of applications.

Taxonomies

Taxonomies are often critical to text analytics. A *taxonomy* is a method for organizing information into hierarchical relationships. It is sometimes referred to as a way of organizing categories. Because a taxonomy defines the relationships between the terms a company uses, it makes it easier to find and then analyze text.

For example, a telecommunications service provider offers both wired and wireless service. Within the wireless service, the company may support cellular phones and Internet access. The company may then have two or more ways of categorizing cellular phone service, such as plans and phone types. The taxonomy could reach all the way down to the parts of a phone itself.

Taxonomies can also use synonyms and alternate expressions, recognizing that cellphone, cellular phone, and mobile phone are all the same. These taxonomies can be quite complex and can take a long while to develop.

Some vendors will state that a taxonomy is not necessary when using their product and that business users can categorize already extracted information. This will actually depend on the subjects you're interested in. Often, the topics can be very complex, nuanced, or specific to a certain industry. That's going to require a focused taxonomy.

Putting Your Results Together with Structured Data

After your unstructured data is structured, you can combine it with other structured information that might exist in your data warehouse, and then apply business intelligence or data-mining tools to gather further insight.

For example, in Table 13-3, text analytics results are merged with structured billing information. You can see that the contents of Table 13-3 are the same as Table 13-1, except we've added a Segment column on the right. Essentially, you can match information from your customers that live in the billing system with the information from the call center notes. Of course, when prospects call in, no information is available to match; this is why "XXXX" appear in these rows.

Table 13-3 Marrying Structured and Unstructured Data

Identifier	*Entity*	*Issue*	*Sentiment*	*Segment*
Cust XYZ	Plan A	Roll-over minutes	Neutral	Gold
Cust ABC	Plan A	Roll-over minutes	Negative	Silver
XXXX	Plan A	Expensive	Neutral	XXX
XXXX	Plan A	Data plan	Neutral	XXX
Cust XYT	Plan A	Data plan	Negative	Bronze

In this example, the structured data together with the unstructured data indicate that at least one of your customers is a gold customer, so it would be worthwhile for the company to make an extra effort to retain him or her. Of course, in reality, you will have a lot more data than this to work with.

Putting Big Data to Use

The wireless promotion use case is just one example of how text analytics can be used to help gain insight into data. So, what if the data is big data? A big data use case would mean that the unstructured data being analyzed is either high volume, high velocity, or both. The following sections describe a few examples.

Voice of the customer

Optimizing the customer experience and improving customer retention are dominant drivers for many service industries. Organizations concerned with these issues might ask questions such as

- What are major areas of complaints by customers and how are these changing over time?
- What is the level of satisfaction of customers with specific services?

- What are the most frequent issues that lead to customer churn?
- What are some key customer segments that provide higher potential upsell opportunities?

Information, such as e-mails to the company, customer satisfaction surveys, call center notes, and other internal documents, hold a lot of information about customer concerns and sentiment. Text analytics can help to identify and address causes of customer dissatisfaction in a timely manner. It can help improve brand image by proactively solving problems before they become a big sticking point with customers.

Is this a big data problem? It can be. It depends on the volume of the information. You may have a large volume of information that is delivered in batch mode. Companies may want to merge this data with structured data, as we discuss earlier in this chapter.

Social media analytics

Another form of voice of the customer or customer experience management, social media analytics, has gotten a lot of visibility recently and, in fact, is helping to drive the text analytics market. In social media analytics, data across the Internet is gathered together. This includes unstructured text from blogs, microblogs, news articles, text from online forums, and so on. This huge stream of data is then analyzed — often using text analytics — to get answers to questions such as

- What are people saying about my brand?
- What do they like about my brand?
- What do they dislike about my brand?
- How does my brand compare to my competitors'?
- How loyal are my customers?

And, social media isn't just being used by marketers concerned about their brand. The government is using it to look for terrorist conversations. Health agencies are using it to identify public health threats worldwide. The list goes on.

This is a big data use case, especially when you can work with a service provider that can assemble all the tweets from Twitter, together with all the other data.

IBM Watson

You may have seen a machine playing and winning *Jeopardy!* a few years ago. That machine is called Watson. IBM Watson is a set of technologies that processes and analyzes massive amounts of both structured and unstructured data in a unique way. Watson can process and analyze information from 200 million books in three seconds. While Watson is very advanced, it uses technologies that are commercially available with some "secret sauce" technologies that IBM Research has either enhanced or developed. It combines software technologies from big data, content and predictive analytics, and industry-specific software to make it work. IBM is working together with the medical industry to develop a Watson for that industry. It is only the first in a series of Watsons that IBM will develop with its partners.

So what is this secret sauce? Watson understands natural language, generates and evaluates hypotheses, and adapts and learns.

First, Watson uses Natural Language Processing. IBM is using a set of annotators to extract information like symptoms, age, location, and so on. Watson is processing vast amounts of this unstructured data quickly, using an architecture designed for this.

Second, Watson works by generating *hypotheses* that are potential answers to a question. It is trained by feeding question-and-answer (Q/A) data into the system. In other words, it is shown representative questions and it learns from the supplied answers. This is called *evidence-based learning*. The goal is to generate a model that can produce a confidence score (think logistic regression with a bunch of attributes). Watson starts with a generic statistical model, then look at the first Q/A, and use that to tweak coefficients. As it gains more evidence, it continues to tweak the coefficients until it can "say" that confidence is high. Training Watson is key because what is really happening is that the trainers are building statistical models that are scored. At the end of the training, Watson has a system that has feature vectors and models so that eventually it can use the model to probabilistically score the answers. The key here is something that *Jeopardy!* did not showcase, which is that it is not deterministic (that is, using rules). Watson is probabilistic and that makes it dynamic.

When Watson generates a hypothesis, it then scores the hypothesis based on the evidence. Its goal is to get the right answer for the right reason. (So, theoretically, if five symptoms must be positive for a certain disease and four must be negative and Watson only has four of the nine pieces of information, it could ask for more.) The hypothesis with the highest score is presented. By the end of the analysis, Watson is confident when it knows the answer and when it doesn't know the answer.

Here's an example. Suppose that you see your doctor because you are not feeling well. Specifically, you might have heart palpitations, fatigue, hair loss, and muscle weakness. You decide to go see a doctor to determine whether something is wrong with your thyroid or whether it is something else. If your doctor has access to a Watson system, he could use it to help advise him regarding your diagnosis. In this case, Watson would already have ingested and curated all the information in books and journals associated with thyroid disease. It also has the diagnosis and related information from other patients from this hospital and other doctors in the practice from the electronic medical records of prior cases that it has in its data banks.

(continued)

(continued)

Based on the first set of symptoms you might report, it would generate a hypothesis along with probabilities associated with the hypothesis (for example, 60 percent hyperthyroidism, 40 percent anxiety, and so on). It might then ask for more information. As it is fed this information, such as patient history, Watson would continue to refine its hypothesis along with the probability of the hypothesis being correct. After it is given all the information and it iterates through it and presents the diagnosis with the highest confidence level, the physician would use this information to help assist him in making the diagnosis and developing a treatment plan. If Watson doesn't know the answer, it will state that it does not have an answer or doesn't have enough information to provide an answer.

IBM likens the process of training a Watson to teaching a child how to learn. A child can read a book to learn. However, he can also learn by a teacher asking questions and reinforcing the answers about that text.

Text Analytics Tools for Big Data

In the following sections, we provide an overview of some of the players in this market. Some are small while others are household names. Some call what they do *big data text analytics,* while some just refer to it as *text analytics.*

Attensity

Attensity (www.attensity.com) is one of the original text analytics companies that began developing and selling products more than ten years ago. At this time, it has over 150 enterprise customers and one of the world's largest NLP development groups. Attensity offers several engines for text analytics. These include Auto-Classification, Entity Extraction, and Exhaustive Extraction. Exhaustive Extraction is Attensity's flagship technology that automatically extracts facts from parsed text (who did what to whom, when, where, under what conditions) and organizes this information.

The company is focused on social and multichannel analytics and engagement by analyzing text for reporting from internal and external sources and then routing it to business users for engagement. It recently purchased Biz360, a social media company that aggregates huge streams of social media. It has developed a grid computing system that provides high-performance capabilities for processing massive amounts of real-time text. Attensity uses a Hadoop framework (MapReduce, HDFS, and HBase) to store data. It also has a data-queuing system that creates an orchestration process that recognizes spikes in inbound data and adjusts processing across more/less servers as needed.

Clarabridge

Another pure-play text analytics vendor, Clarabridge (www.clarabridge.com) is actually a spin-off of a business intelligence (BI) consulting firm (called Claraview) that realized the need to deal with unstructured data. Its goal is to help companies drive measurable business value by looking at the customer holistically, pinpointing key experiences and issues, and helping everyone in an organization take actions and collaborate in real time. This includes real-time determination of sentiment and classification of customer feedback data / text and staging the verbatim for future processing into the Clarabridge system.

At this time, Clarabridge is offering its customers some sophisticated and interesting features, including single-click root cause analysis to identify what is causing a change in the volume of text feeds, sentiment, or satisfaction associated with emerging issues. It also offers its solution as a Software as a Service (SaaS).

IBM

Software giant IBM (www.ibm.com) offers several solutions in the text analytics space under its Smarter Planet strategy umbrella. Aside from Watson and IBM SPSS (see Chapter 12 for more on SPSS), IBM also offers IBM Content Analytics with Enterprise Search (ICAES). IBM Content Analytics was developed based on work done at IBM Research.

IBM Content Analytics is used to transform content into analyzed information, and this is available for detailed analyses similar to the way structured data would be analyzed in a BI toolset. IBM Content Analytics and Enterprise Search were once two separate products. The converged solution targets both enhanced enterprise search that uses text analytics, as well as standalone content analytics needs. ICAES has tight integration with the IBM InfoSphere BigInsights platform, enabling very large search and content analytics collections.

OpenText

OpenText (www.opentext.com), a Canadian-based company, is probably best known for its leadership in enterprise information management (EIM) solutions. Its vision revolves around managing, securing, and extracting value from the unstructured data of enterprises. It provides what it terms "semantic middleware." According to the company, its semantic technology evolution is rooted in its capability "to enable real-time analytics with high accuracy on large data sets (that is, content) across languages, formats, and industry domains." The idea behind semantic middleware is that semantics can be exposed at different levels and work with different technologies (for example, document management, predictive analytics, and so on) to address business issues. In other words, the text analytics can be enabled and utilized where needed. OpenText provides this middleware as a stand-alone product to be used in a variety of solutions as well as embedded in its products.

REMEMBER

The idea of pluggable semantic enablers is starting to gain more steam, and smaller players are also looking at ways that these enablers can provide value to big data applications.

SAS

SAS (`www.sas.com`) has been solving complex big data problems for a long time. Several years ago, it purchased text analytics vendor Teragram to enhance its strategy to use both structured and unstructured data in analysis and to integrate this data for descriptive and predictive modeling. Now, its text analytics capabilities are part of its overall analytics platform and text data is viewed as simply another source of data.

SAS continues to innovate in the area of high-performance analytics to ensure that performance meets customer expectations. The goal is to take problems that used to take weeks to solve and solve them in days, or problems that used to take days to solve and solve them in minutes instead. For example, the SAS High Performance Analytics Server is an in-memory solution that allows you to develop analytical models using complete data, not just a subset of aggregate data. SAS says that you can use thousands of variables and millions of documents as part of this analysis. The solution runs on EMC Greenplum or Teradata appliances as well as on commodity hardware using Hadoop Distributed File System (HDFS).

Chapter 14

Customized Approaches for Analysis of Big Data

In This Chapter

- New models and approaches evolving to support big data analysis
- Full custom versus semi-custom analysis approaches
- Optimal environment for big data analysis
- Big to small is the goal

The beauty of big data is that, theoretically, all the data you need, both inside and outside of your company, can be used to drive your analysis. Ideally, this means that if you increase the amount or type of data you analyze, you can derive new insights from it.

As we discuss in Chapters 12 and 13, many tools used to analyze big data are an evolution of what's already out there in the market in terms of business intelligence and advanced analysis. These include data-mining software, predictive modeling, advanced statistics, and text analytics. As we also mention in the previous two chapters, often, vendors have had to rewrite their algorithms to run this software across new big data infrastructures.

Figure 14-1 shows the focus of this chapter.

According to some experts in the field, the mind-set around analyzing big data is different than traditional analysis and is one of exploration and experimentation — going where the data takes you. While others disagree, the reality is that the big data analytics ecosystem will require some new technology platforms, algorithms, and skill sets to support this kind of analysis — especially when it comes to pushing the envelope in terms of what can be done. Because we are in the early stages of big data usage and adoption, a large percentage of the analysis will need to be delivered in the form of "customized" or "special-purpose" applications. This chapter examines some of these changes and describes how to address them.

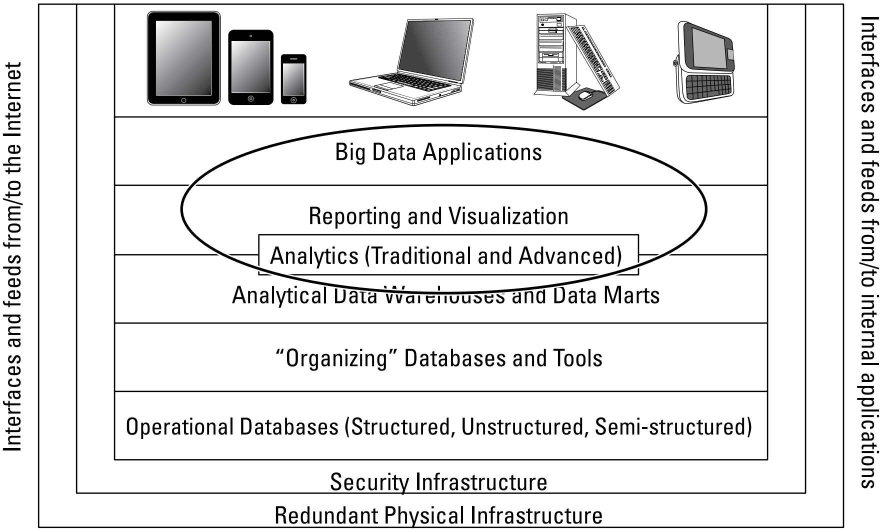

Interfaces and feeds from/to the Internet
Big Data Applications
Reporting and Visualization
Analytics (Traditional and Advanced)
Analytical Data Warehouses and Data Marts
"Organizing" Databases and Tools
Operational Databases (Structured, Unstructured, Semi-structured)
Security Infrastructure
Redundant Physical Infrastructure
Interfaces and feeds from/to internal applications

Figure 14-1: Layers of the big data reference architecture needed for customer analysis.

Building New Models and Approaches to Support Big Data

Big data analysis has gotten a lot of hype recently, and for good reason. Companies are excited to be able to access and analyze data that they've been collecting or want to gain insight from, but have not been able to manage or analyze effectively. These companies know that something is out there, but until recently, have not been able to mine it. This pushing the envelope on analysis is an exciting aspect of the big data analysis movement. It might involve visualizing huge amounts of disparate data, or it might involve advanced analyzed streaming at you in real time. It is evolutionary in some respects and revolutionary in others.

Characteristics of big data analysis

So, what's different when your company is pushing the envelope with big data analysis? We talk a little bit about this in Chapter 12. We describe that the infrastructure supporting big data analysis is different and algorithms have been changed to be infrastructure aware.

Big data analysis should be viewed from two perspectives:

- Decision-oriented
- Action-oriented

Decision-oriented analysis is more akin to traditional business intelligence. We look at selective subsets and representations of larger data sources and try to apply the results to the process of making business decisions. Certainly these decisions might result in some kind of action or process change, but the purpose of the analysis is to augment decision making.

Action-oriented analysis is used for rapid response, when a pattern emerges or specific kinds of data are detected and action is required. We discuss these kinds of use cases throughout the book, but here is where the "rubber meets the road." Taking advantage of big data through analysis and causing proactive or reactive behavior changes offer great potential for early adopters.

Finding and utilizing big data by creating analysis applications can hold the key to extracting value sooner rather than later. To accomplish this task, it is more effective to build these custom applications from scratch or by leveraging platforms and/or components. We cover this topic later in this chapter.

First, we look at some of the additional characteristics of big data analysis that make it different from traditional kinds of analysis aside from the three Vs of volume, velocity, and variety:

- **It can be *programmatic.*** One of the biggest changes in terms of analysis is that in the past you were dealing with data sets you could manually load into an application and visualize and explore. With big data analysis, you may be faced with a situation where you might start with the raw data that often needs to be handled *programmatically* (using code) to manipulate it or to do any kind of exploration because of the scale of the data.
- **It can be *data driven.*** While many data scientists use a hypothesis-driven approach to data analysis (develop a premise and collect data to see whether that premise is correct), you can also use the data to drive the analysis — especially if you've collected huge amounts of it. For example, you can use a machine-learning algorithm (for more on machine learning, see Chapter 12) to do this kind of hypothesis-free analysis.
- **It can use a lot of *attributes.*** In the past, you might have been dealing with hundreds of attributes or characteristics of that data source. Now you might be dealing with hundreds of gigabytes of data that consist of thousands of attributes and millions of observations. Everything is now happening on a larger scale.
- **It can be *iterative.*** More compute power means that you can iterate on your models until you get them the way you want them. Here's an example. Assume that you're building a model that is trying to find the predictors for certain customer behaviors associated with certain products. You might start off extracting a reasonable sample of data or connecting to where the data resides. You might build a model to test a hypothesis.

Whereas in the past you might not have had that much memory to make your model work effectively, you will need a tremendous amount of physical memory to go through the necessary iterations required to train the algorithm. It may also be necessary to use advanced computing techniques like natural language processing or neural networks that automatically evolve the model based on learning as more data is added.

- **It can be *quick* to get the compute cycles you need by leveraging a cloud-based Infrastructure as a Service.** With Infrastructure as a Service (IaaS) platforms like Amazon Cloud Services (ACS), you can rapidly provision a cluster of machines to ingest large data sets and analyze them quickly.

Now that you have a better understanding of some of the characteristics, look at some of the means at your disposal for analyzing big data.

Understanding Different Approaches to Big Data Analysis

In many cases, big data analysis will be represented to the end user through reports and visualizations. Because the raw data can be incomprehensively varied, you will have to rely on analysis tools and techniques to help present the data in meaningful ways. Traditionally generated reports are familiar, but they may not be able to provide new insights or create the unanticipated findings decision makers are searching for. Data visualization techniques will help, but they too will need to be enhanced or supported by more sophisticated tools to address big data.

While traditional reporting and visualization are familiar, they are insufficient, so it will become necessary to create new applications and approaches for analysis of big data. Otherwise, you will be in a holding pattern until vendors begin to catch up with the demand. Even when they catch up, the resulting solution may not do what you need. Early adoption of big data requires the creation of new applications designed to address analysis requirements and time frames. Why is this so important? It is important because a well-used representation from traditional data analysis will be inadequate.

These new applications will fall broadly into two categories: custom (coded from scratch) or semi-custom (based on frameworks or components). We examine some examples to help understand why and how we can use these approaches to make big data more useful in our daily work lives sooner rather than later.

Custom applications for big data analysis

In general, a custom application is created for a specific purpose or a related set of purposes. Certain areas of a business or organization will always require a custom set of technologies to support unique activities or to provide a competitive advantage. For example, if you are involved in financial services, you want your trading applications to be faster and more accurate than your competitors'. In contrast, the applications that do your client billing probably do not need very much specialization, so a packaged system can do the trick.

For big data analysis, the purpose of custom application development is to speed the time to decision or the time to action. As big data evolves as a science and a market, software vendors of traditional solutions will be slow to bring new technologies to market. Little value exists in a big data infrastructure if very few opportunities are available to decide or act upon because of the lack of analysis capabilities germane to the business area. As we discuss in Chapters 12 and 13, some packages support a wide variety of analysis techniques for big data. The vendors discussed in these chapters can utilize their technology components to help build solutions for their customers. However, the reality is that there is no such thing as a completely packaged application that will work out of the box for a sophisticated big data solution. We now examine some additional options that are available for those of us who may need custom analysis applications for big data.

R environment

The "R" environment is based on the "S" statistics and analysis language developed in the 1990s by Bell Laboratories. It is maintained by the GNU project and is available under the GNU license. Over the years, many users of S and R have contributed greatly to the base system, enhancing and expanding its capabilities. While challenging to fully comprehend, its depth and flexibility make it a compelling choice for analytics application developers and "power users." In addition, the CRAN (Comprehensive R Archive Network) R project maintains a worldwide set of File Transfer Protocol (FTP) and web servers with the most up-to-date versions of the R environment. A commercially supported, enterprise version of R is also available from Revolution Analytics in Palo Alto, California (`www.revolution-computing.com`).

More specifically, R is an integrated suite of software tools and technologies designed to create custom applications used to facilitate data manipulation, calculation, analysis, and visual display. Among other advanced capabilities, it supports

- Effective data-handling and manipulation components.
- Operators for calculations on arrays and other types of ordered data.
- Tools specific to a wide variety of data analyses.

- Advanced visualization capabilities.
- S programming language designed by programmers, for programmers with many familiar constructs, including conditionals, loops, user-defined recursive functions, and a broad range of input and output facilities. Most of the system-supplied functions are written in the S language.

R is a vehicle for developing new methods of interactive big data analysis. It has developed rapidly and has been extended by a large collection of *packages.* It is well suited to single-use, custom applications for analysis of big data sources.

Google Prediction API

The Google Prediction API is an example of an emerging class of big data analysis application tools. It is available on the Google developers website and is well documented and provided with several mechanisms for access using different programming languages. To help you get started, it is freely available (with some restrictions) for six months. Subsequent licensing is very modest and project based.

The Prediction API is fairly simple. It looks for patterns and matches them to proscriptive, prescriptive, or other existing patterns. While performing its pattern matching, it also "learns." In other words, the more you use it, the smarter it gets. What kinds of things could you "learn" from using the Prediction API? Suppose that you wanted to understand consumer behavior. You might want to source postings from Facebook, Twitter, Amazon, and/or foursquare social sites looking for specific patterns of behavior. If you are a consumer products company, you might want to suggest new or existing products based on the information on the social sites. If you are a Hollywood production company, you might want to notify people of a new movie with one of their favorite stars. The Prediction API gives you the opportunity to predict (or even encourage) future behaviors by analyzing habits and prior actions.

Prediction is implemented as a RESTful API with language support for .NET, Java, PHP, JavaScript, Python, Ruby, and many others. Google also provides scripts for accessing the API as well as a client library for R.

Predictive analysis is one of the most powerful potential capabilities of big data, and the Google Prediction API is a very useful tool for creating custom applications.

As big data evolves, many new types of custom application tools will be introduced to the market. Some may resemble R, and others (like Google Prediction API) will be introduced as APIs or libraries that programmers can use to create new ways to compute and analyze big data. In the real world, many people do not have software developers available to code custom applications. Fortunately, some other means are available and emerging that you can use to address the needs of analysis users.

Semi-custom applications for big data analysis

In truth, what many people perceive as custom applications are actually created using “packaged” or third-party components like libraries. It is not always necessary to completely code a new application. (When it is necessary, no substitute exists.) Using packaged applications or components requires developers or analysts to write code to “knit together” these components into a working custom application. The following are reasons why this is a sound approach:

- **Speed to deployment:** Because you don’t have to write every part of the application, the development time can be greatly reduced.
- **Stability:** Using well-constructed, reliable, third-party components can help to make the custom application more resilient.
- **Better quality:** Packaged components are often subject to higher quality standards because they are deployed into a wide variety of environments and domains.
- **More flexibility:** If a better component comes along, it can be swapped into the application, extending the lifetime, adaptability, and usefulness of the custom application.

Another type of semi-custom application is one where the source code is available and is modified for a particular purpose. This can be an efficient approach because there are quite a few examples of application building blocks available to incorporate into your semi-custom application. Some of these include:

- **TA-Lib:** The Technical Analysis library is used extensively by software developers who need to perform technical analysis of financial market data. It is available as open source under the BSD license, allowing it to be integrated into semi-custom applications.
- **JUNG:** The Java Universal Network Graph framework is a library that provides a common framework for analysis and visualization of data that can be represented by a graph or network. It is useful for social network analysis, importance measures (PageRank, hits), and data mining. It is available as open source under the BSD license.
- **GeoTools:** An open source geospatial toolkit for manipulating GIS data in many forms, analyzing spatial and non-spatial attributes or GIS data, and creating graphs and networks of the data. It is available under the GPL2 license, allowing for integration into semi-custom applications.

Going mobile

It is true that many (if not all) mobile applications are custom. Some third-party package providers offer mobile access, often through a mobile application, but they are generally not useful outside the providers' interests. As a result, many of the emerging custom component developers are delivering technology that can help create mobile applications for big data more easily.

The velocity of big data, coupled with its variety, will cause a move toward real-time observations, allowing better decision making or quick action. As the market evolves, it is likely that most of these observations will be the result of custom applications designed to augment the ability to react to changes in the environment. Analysis frameworks and components will help to create, modify, share, and maintain these applications with greater ease and efficiency.

Characteristics of a Big Data Analysis Framework

Even though new sets of tools continue to be available to help you manage and analyze big data more effectively, you may not be able to get what you need from what's already out there. In addition, a range of technologies that we talk about earlier in this book can support big data analysis and also support requirements such as availability, scalability, and high performance. Some of these technologies include big data appliances, columnar databases, in-memory databases, nonrelational databases, and massively parallel processing engines. Chapters 1, 4, and 7 cover these topics in more detail.

So, what are business users looking for when it comes to big data analysis? The answer to that question depends on the type of business problem they are trying to solve. Earlier in the chapter, we discuss decision orientation and action orientation as two broad types of business challenges. Many of the characteristics are common to both, and because decisions often lead to actions, the commonality is required. Some important considerations you need to take in as you select a big data application analysis framework include the following:

- **Support for multiple data types:** Many organizations are incorporating, or expect to incorporate, all types of data as part of their big data deployments, including structured, semi-structured, and unstructured data.

- **Handle batch processing and/or real time data streams:** Action orientation is a product of analysis on real-time data streams, while decision orientation can be adequately served by batch processing. Some users will require both, as they evolve to include varying forms of analysis.
- **Utilize what already exists in your environment:** To get the right context, it may be important to leverage existing data and algorithms in the big data analysis framework.
- **Support NoSQL and other newer forms of accessing data:** While organizations will continue to use SQL, many are also looking at newer forms of data access to support faster response times or faster times to decision.
- **Overcome low latency:** If you're going to be dealing with high data velocity, you're going to need a framework that can support the requirements for speed and performance.
- **Provide cheap storage:** Big data means potentially lots of storage — depending on how much data you want to process and/or keep. This means that storage management and the resultant storage costs are important considerations.
- **Integrate with cloud deployments:** The cloud can provide storage and compute capacity on demand. More and more companies are using the cloud as an analysis "sandbox." Increasingly, the cloud is becoming an important deployment model to integrate existing systems with cloud deployments (either public or private) in a hybrid model. In addition, big data cloud services are beginning to emerge that will benefit customers. For more on this issue, check out Chapter 11.

While all these characteristics are important, the perceived and actual value of creating applications from a framework is quicker time to deployment. With all these capabilities in mind, we look at an example of a big data analysis application framework from a company called Continuity.

The Continuity AppFabric (`www.continuity.com`) is a framework supporting the development and deployment of big data applications. Deployment can be as a single instance, private cloud, or public cloud, without any recoding required for the target environment. The AppFabric itself is a set of technologies specifically designed to abstract away the vagaries of low-level big data technologies. The application builder is an Eclipse plug-in permitting the developer to build, test, and debug locally and in familiar surroundings.

AppFabric capabilities include the following:

- Stream support for real-time analysis and reaction
- Unified API, eliminating the need to write to big data infrastructures
- Query interfaces for simple results and support for pluggable query processors

- Data sets representing queryable data and tables accessible from the Unified API
- Reading and writing of data independent of input or output formats or underlying component specifics (such as Hadoop data operations)
- Transaction-based event processing
- Multimodal deployment to a single node or the cloud

This approach is going to gain traction for big data application development primarily because of the plethora of tools and technologies required to create a big data environment. If a developer can write to a higher-level API, requiring that the "fabric" or abstraction layer manage the specifics of the underlying components, you should expect high-quality, reliable applications that can be easily modified and deployed.

It is a rare company that can afford to build a big data analysis capability from scratch. Therefore, it is best to think about your big data deployment as an ecosystem of people, processes, and technologies. Big data analysis is not an island. It is connected to many other data environments and business process environments throughout your enterprise. Even though these are the early days in the big data movement and many projects are experimenting with big data analysis in isolation from their overall computing environment, you need to think about integration as a requirement. As big data analysis becomes more mainstream, it should remain isolated from the rest of the data management environment.

Software developers seldom work in isolation. Likewise, data scientists and analysis experts like to share discoveries and leverage existing assets. The need to collaborate and share is even more pronounced in an emerging technology area. In fact, the lack of collaboration can be costly in many ways. Large organizations can benefit from tools that drive collaborations. Very often people doing similar work are unaware of each other's efforts leading to duplicate work (or worse!). This is costly in terms of money and productivity. Jump-starting a project with existing solutions can make a difference in quality and time-to-market.

Another good example of an application framework is OpenChorus (`www.openchorus.org`). In addition to rapid development of big data analysis applications, it also supports collaboration and provides many other features important to software developers, like tool integration, version control, and configuration management.

Open Chorus is a project maintained by EMC Corporation and is available under the Apache 2.0 license. EMC also produces and supports a commercial version of Chorus. Both Open Chorus and Chorus have vibrant partner networks as well as a large set of individual and corporate contributors.

Open Chorus is a generic framework. Its leading feature is the capability to create a communal "hub" for sharing big data sources, insights, analysis techniques, and visualizations. Open Chorus provides the following:

- Repository of analysis tools, artifacts, and techniques with complete versioning, change tracking, and archiving
- Workspaces and sandboxes that are self-provisioned and easily maintained by community members
- Visualizations, including heat maps, time series, histograms, and so on
- Federated search of any and all data assets, including Hadoop, metadata, SQL repositories, and comments
- Collaboration through social networking–like features encouraging discovery, sharing, and brainstorming
- Extensibility for integration of third-party components and technologies

As big data evolves, you will see the introduction of new kinds of application frameworks. Many of these will support mobile application development, while others will address vertical application areas. In any case, they are an important tool for early adopters of big data.

Big to Small: A Big Data Paradox

You'll find a nuance about big data analysis. It's really about small data. While this may seem confusing and counter to the whole premise of this book, small data is the product of big data analysis. This is not a new concept, nor is it unfamiliar to people who have been doing data analysis for any length of time. The overall working space is larger, but the answers lie somewhere in the "small."

Traditional data analysis began with databases filled with customer information, product information, transactions, telemetry data, and so on. Even then, too much data was available to efficiently analyze. Systems, networks, and software didn't have the performance or capacity to address the scale. As an industry, we addressed the shortcomings by creating smaller data sets.

These smaller data sets were still fairly substantive, and we quickly discovered other shortcomings; the most glaring was the mismatch between the data and the working context. If you worked in Accounts Payable, you had to look at a large amount of unrelated data to do your job. Again, the industry responded by creating smaller, contextually relevant data sets — big to small to smaller still.

You may recognize this as the migration from databases to data warehouses to data marts. More often than not, the data for the warehouses and the marts was chosen on arbitrary or experimental parameters resulting in a great deal of trial and error. We weren't getting the perspectives we needed or were possible because the capacity reductions weren't based on computational fact.

Enter big data, with all its volumes, velocities, and varieties, and the problem remains or perhaps worsens. We have addressed the shortcomings of the infrastructure and can store and process huge amounts of additional data, but we also had to introduce new technologies specifically to help us manage big data.

Despite the outward appearances, this is a wonderful thing. Today and in the future, we will have more data than we can imagine and we'll have the means to capture and manage it. What is more necessary than ever is the capability to analyze the *right* data in a timely enough fashion to make decisions and take actions. We will still shrink the data sets into "fighting trim," but we can do so computationally. We process the big data and turn it into small data so that it's easier to comprehend. It's more precise and, because it was derived from a much larger starting point, it's more contextually relevant.