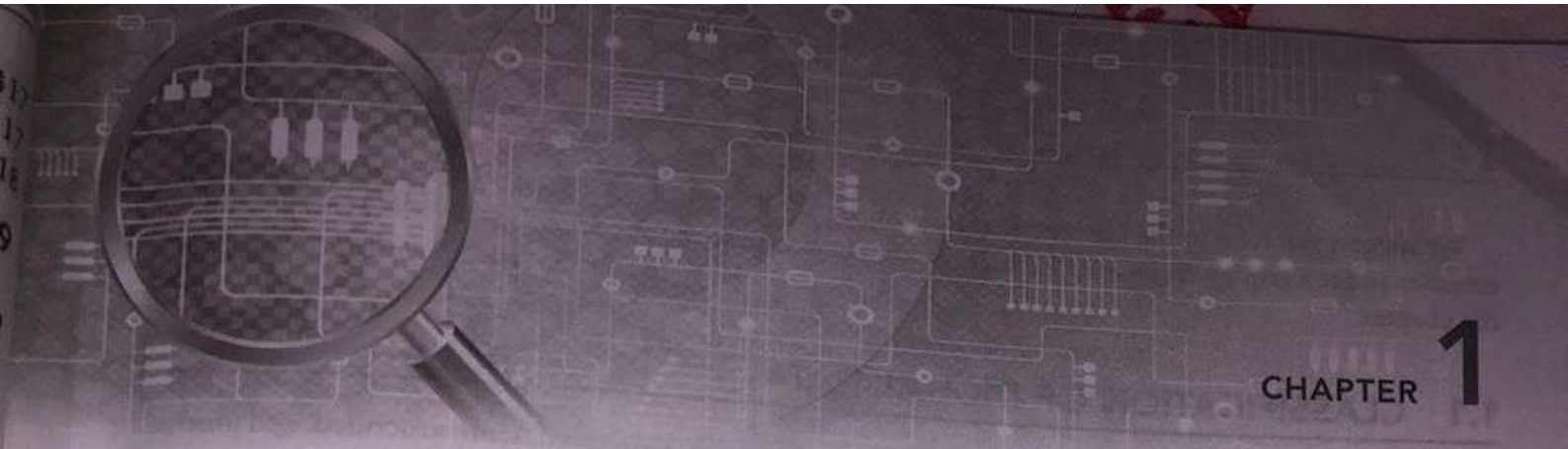

CHAPTER 1

Types of Digital Data

BRIEF CONTENTS

"In God we trust, all others must bring data."

– W. Edwards Deming

WHAT'S IN STORE?

Irrespective of the size of the enterprise (big or small), data continues to be a precious and irreplaceable asset. Data is present internal to the enterprise and also exists outside the four walls and firewalls of the enterprise. Data is present in homogeneous sources as well as in heterogeneous sources. The need of the hour is to understand, manage, process, and take the data for analysis to draw valuable insights.

Data → Information

Information → Insights

This chapter is a "must read" for first-time learners interested in understanding the role of data in business intelligence and business analysis and businesses at large. This chapter will introduce you to the various formats of digital data (structured, semi-structured, and unstructured data), the sources of each format of data, the issues with the terminology of unstructured data, etc.

We suggest you refer to the learning resources suggested at the end of this chapter and also attempt all the exercises to get a grip on this topic. We suggest you make your own notes/bookmarks while reading through the chapter.

1.1 CLASSIFICATION OF DIGITAL DATA

As depicted in Figure 1.1, digital data can be broadly classified into structured, semi-structured, and unstructured data.

1. **Unstructured data:** This is the data which does not conform to a data model or is not in a form which can be used easily by a computer program. About 80–90% data of an organization is in this format; for example, memos, chat rooms, PowerPoint presentations, images, videos, letters, researches, white papers, body of an email, etc.
2. **Semi-structured data:** This is the data which does not conform to a data model but has some structure. However, it is not in a form which can be used easily by a computer program; for example, emails, XML, markup languages like HTML, etc. Metadata for this data is available but is not sufficient.
3. **Structured data:** This is the data which is in an organized form (e.g., in rows and columns) and can be easily used by a computer program. Relationships exist between entities of data, such as classes and their objects. Data stored in databases is an example of structured data.

Ever since the 1980s most of the enterprise data has been stored in relational databases complete with rows/records/tuples, columns/attributes/fields, primary keys, foreign keys, etc. Over a period of time Relational Database Management System (RDBMS) matured and the RDBMS, as they are available today, have become more robust, cost-effective, and efficient. We have grown comfortable working with RDBMS – the storage, retrieval, and management of data has been immensely simplified. The data held in RDBMS is typically structured data. However, with the Internet connecting the world, data that existed beyond one's enterprise started to become an integral part of daily transactions. This data grew by leaps and bounds so much so that it became difficult for the enterprises to ignore it. All of this data was not structured. A lot of it was unstructured. In fact, Gartner estimates that almost 80% of data generated in any enterprise today is unstructured data. Roughly around 10% of data is in the structured and semi-structured category. Refer Figure 1.2.

1.1.1 Structured Data

Let us begin with a very basic question – When do we say that the data is structured? The simple answer is when data conforms to a pre-defined schema/structure we say it is structured data.

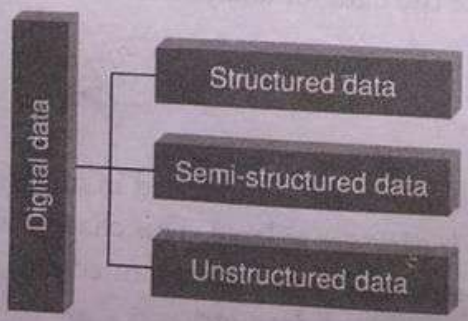

Figure 1.1 Classification of digital data.

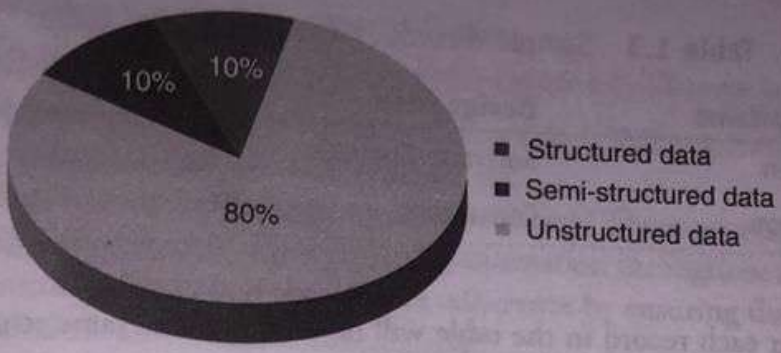

Figure 1.2 Approximate percentage distribution of digital data.

Think structured data, and think data model – a model of the types of business data that we intend to store, process, and access. Let us discuss this in the context of an RDBMS. Most of the structured data is held in RDBMS. An RDBMS conforms to the relational data model wherein the data is stored in rows/columns. Refer Table 1.1.

The number of rows/records/tuples in a relation is called the *cardinality of a relation* and the number of columns is referred to as the *degree of a relation*.

The first step is the design of a relation/table, the fields/columns to store the data, the type of data that will be stored [number (integer or real), alphabets, date, Boolean, etc.]. Next we think of the constraints that we would like our data to conform to (constraints such as UNIQUE values in the column, NOT NULL values in the column, a business constraint such as the value held in the column should not drop below 50, the set of permissible values in the column such as the column should accept only "CS", "IS", "MS", etc., as input).

To explain further, let us design a table/relation structure to store the details of the employees of an enterprise. Table 1.2 shows the structure/schema of an "Employee" table in a RDBMS such as Oracle.

Table 1.2 is an example of a good structured table (complete with table name, meaningful column names with data types, data length, and the relevant constraints) with absolute adherence to relational data model.

Table 1.1 A relation/table with rows and columns

	Column 1	Column 2	Column 3	Column 4
Row 1				

Table 1.2 Schema of an "Employee" table in a RDBMS such as Oracle

Column Name	Data Type	Constraints
EmpNo	Varchar(10)	PRIMARY KEY
EmpName	Varchar(50)	
Designation	Varchar(25)	NOT NULL
DeptNo	Varchar(5)	
ContactNo	Varchar(10)	NOT NULL

Table 1.3 Sample records in the "Employee" table

EmpNo	EmpName	Designation	DeptNo	ContactNo
E101	Allen	Software Engineer	D1	0999999999
E102	Simon	Consultant	D1	0777777777

It goes without saying that each record in the table will have exactly the same structure. Let us take a look at a few records in Table 1.3.

The tables in an RDBMS can also be related. For example, the above "Employee" table is related to the "Department" table on the basis of the common column, "DeptNo". It is not mandatory for the two tables that are related to have exactly the same name for the common column. On the contrary, the two tables are related on the basis of values held within the column, "DeptNo". Given in Figure 1.3 is a depiction of referential integrity constraint (primary – foreign key) with the "Department" table being the referenced table and "Employee" table being the referencing table.

1.1.1.1 Sources of Structured Data

If your data is highly structured, one can look at leveraging any of the available RDBMS [Oracle Corp. – Oracle, IBM – DB2, Microsoft – Microsoft SQL Server, EMC – Greenplum, Teradata – Teradata, MySQL (open source), PostgreSQL (advanced open source), etc.] to house it. Refer Figure 1.4. These databases are typically used to hold transaction/operational data generated and collected by day-to-day business activities. In other words, the data of the On-Line Transaction Processing (OLTP) systems are generally quite structured.

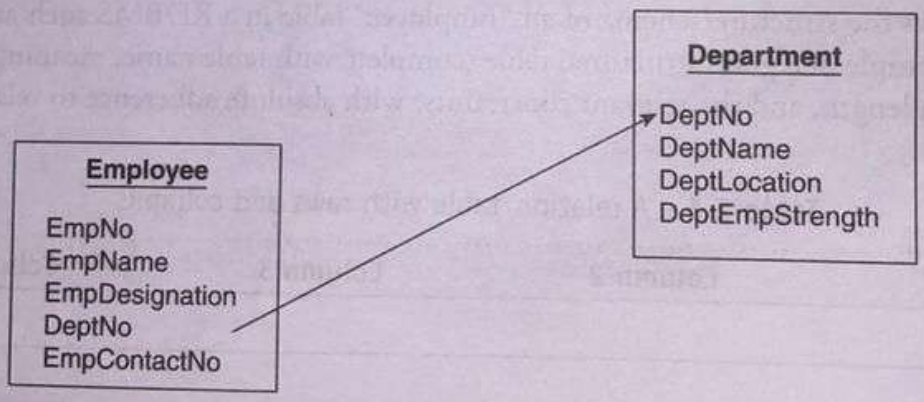

Figure 1.3 Relationship between "Employee" and "Department" tables.

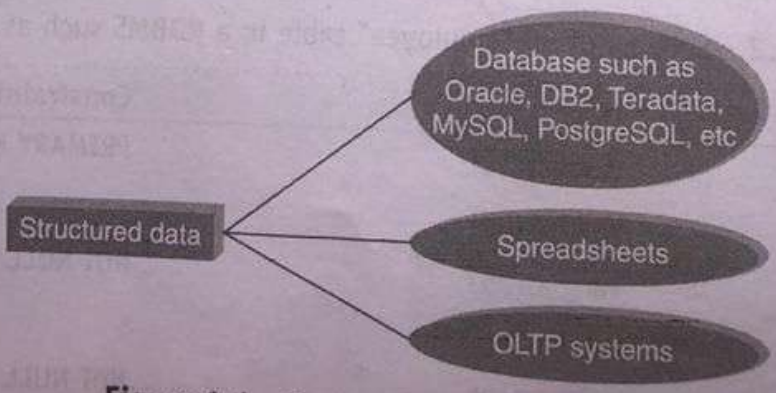

Figure 1.4 Sources of structured data.

1.1.1.2 Ease of Working with Structured Data

Structured data provides the ease of working with it. Refer Figure 1.5. The ease is with respect to the following:

1. **Insert/update/delete:** The Data Manipulation Language (DML) operations provide the required ease with data input, storage, access, process, analysis, etc.
2. **Security:** How does one ensure the security of information? There are available staunch encryption and tokenization solutions to warrant the security of information throughout its lifecycle. Organizations are able to retain control and maintain compliance adherence by ensuring that only authorized individuals are able to decrypt and view sensitive information.
3. **Indexing:** An index is a data structure that speeds up the data retrieval operations (primarily the SELECT DML statement) at the cost of additional writes and storage space, but the benefits that ensue in search operation are worth the additional writes and storage space.
4. **Scalability:** The storage and processing capabilities of the traditional RDBMS can be easily scaled up by increasing the horsepower of the database server (increasing the primary and secondary or peripheral storage capacity, processing capacity of the processor, etc.).
5. **Transaction processing:** RDBMS has support for Atomicity, Consistency, Isolation, and Durability (ACID) properties of transaction. Given next is a quick explanation of the ACID properties:
   - ***Atomicity:*** A transaction is atomic, means that either it happens in its entirety or none of it at all.
   - ***Consistency:*** The database moves from one consistent state to another consistent state. In other words, if the same piece of information is stored at two or more places, they are in complete agreement.
   - ***Isolation:*** The resource allocation to the transaction happens such that the transaction gets the impression that it is the only transaction happening in isolation.
   - ***Durability:*** All changes made to the database during a transaction are permanent and that accounts for the durability of the transaction.

1.1.2 Semi-Structured Data

Semi-structured data is also referred to as self-describing structure. Refer Figure 1.6. It has the following features:

1. It does not conform to the data models that one typically associates with relational databases or any other form of data tables.
2. It uses tags to segregate semantic elements.

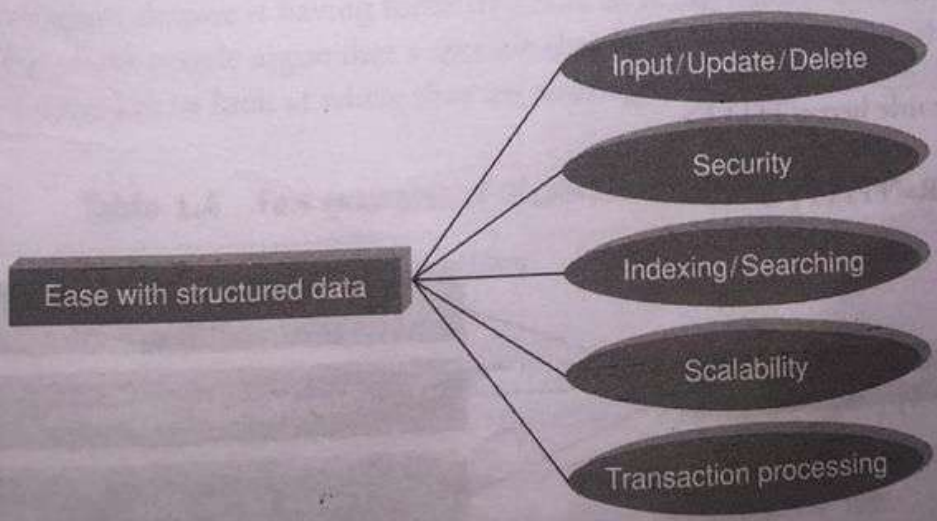

Figure 1.5 Ease of working with structured data.

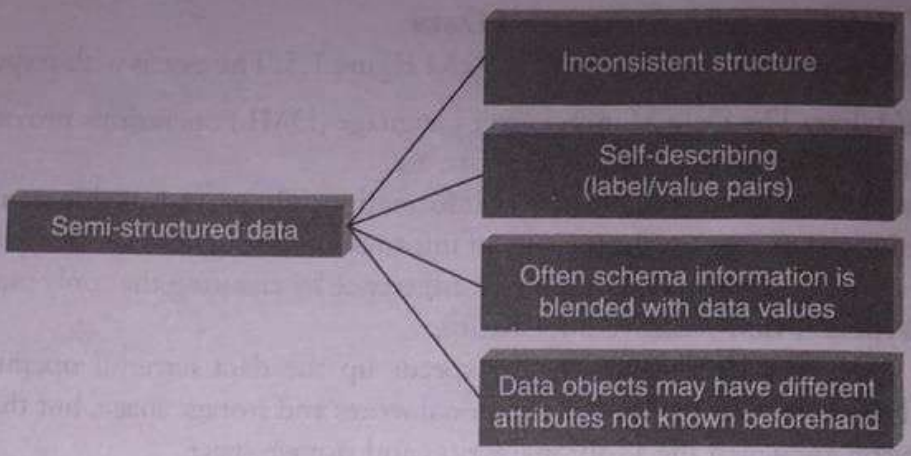

Figure 1.6 Characteristics of semi-structured data.

3. Tags are also used to enforce hierarchies of records and fields within data.
4. There is no separation between the data and the schema. The amount of structure used is dictated by the purpose at hand.
5. In semi-structured data, entities belonging to the same class and also grouped together need not necessarily have the same set of attributes. And if at all, they have the same set of attributes, the order of attributes may not be similar and for all practical purposes it is not important as well.

1.1.2.1 Sources of Semi-Structured Data

Amongst the sources for semi-structured data, the front runners are "XML" and "JSON" as depicted in Figure 1.7.

1. **XML:** eXtensible Markup Language (XML) is hugely popularized by web services developed utilizing the Simple Object Access Protocol (SOAP) principles.
2. **JSON:** Java Script Object Notation (JSON) is used to transmit data between a server and a web application. JSON is popularized by web services developed utilizing the Representational State Transfer (REST) – an architecture style for creating scalable web services. MongoDB (open-source, distributed, NoSQL, documented-oriented database) and Couchbase (originally known as Membase, open-source, distributed, NoSQL, document-oriented database) store data natively in JSON format.

An example of HTML is as follows:

```
<HTML>
<HEAD>
<TITLE>Place your title here</TITLE>
</HEAD>
<BODY BGCOLOR="FFFFFF">
```

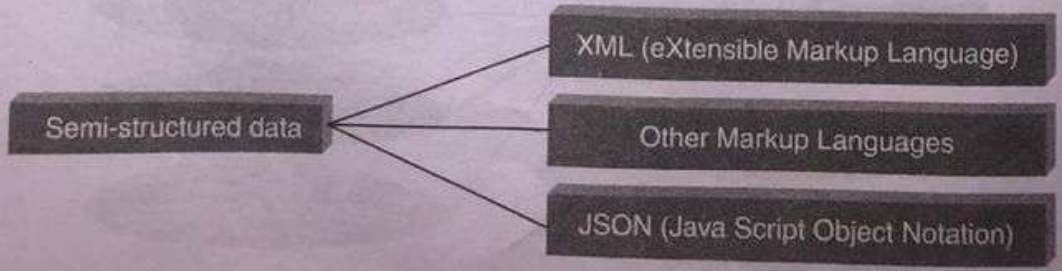

Figure 1.7 Sources of semi-structured data.

```
<CENTER><IMG SRC="clouds.jpg" ALIGN="BOTTOM"></CENTER>
<HR>
<a href="http://bigdatauniversity.com">Link Name</a>
<H1>this is a Header</H1>
<H2>this is a sub Header</H2>
Send me mail at <a href="mailto:support@yourcompany.com">
support@yourcompany.com</a>.
<P>a new paragraph!
<P><B>a new paragraph!</B>
<BR><B><I>this is a new sentence without a paragraph break, in bold italics.</I></B>
<HR>
</BODY>
</HTML>
```

Sample JSON document

```
{
_id:9,
BookTitle: "Fundamentals of Business Analytics",
AuthorName: "Seema Acharya",
Publisher: "Wiley India",
YearofPublication: "2011"
}
```

1.1.3 Unstructured Data

Unstructured data does not conform to any pre-defined data model. In fact, to explain things a little more, let us take a closer look at the various kinds of text available and the possible structure associated with it. As can be seen from the examples quoted in Table 1.4, the structure is quite unpredictable. In Figure 1.8 we look at the other sources of unstructured data.

1.1.3.1 Issues with "Unstructured" Data

Although unstructured data is known NOT to conform to a pre-defined data model or be organized in a pre-defined manner, there are incidents wherein the structure of the data (placed in the unstructured category) can still be implied. As mentioned in Figure 1.9, there could be few other reasons behind placing data in the unstructured category despite it having some structure or being highly structured.

There are situations where people argue that a text file should be in the category of semi-structured data and not unstructured data. Let us look at where they are coming from. Well, the text file does have a name,

Table 1.4 Few examples of disparate unstructured data

Twitter message	Feeling miffed ☹. Victim of twishing.
Facebook post	LOL. C ya. BFN
Log files	127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I; Nav)"
Email	Hey Joan, possible to send across the first cut on the Hadoop chapter by Friday EOD or maybe we can meet up over a cup of coffee. Best regards, Tom

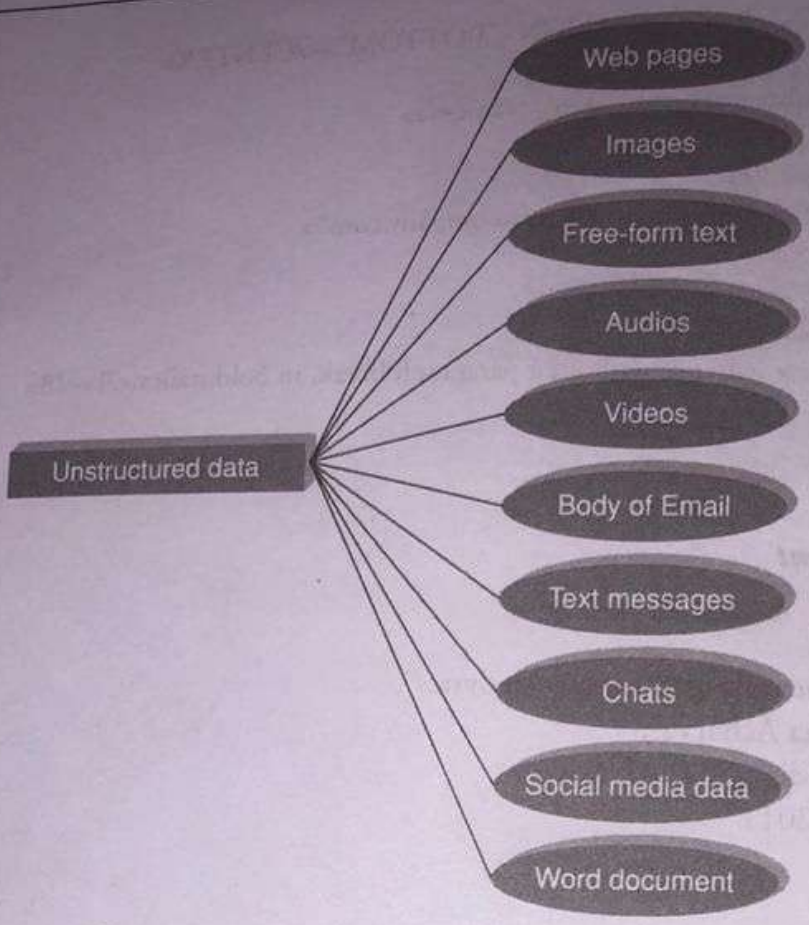

Figure 1.8 Sources of unstructured data.

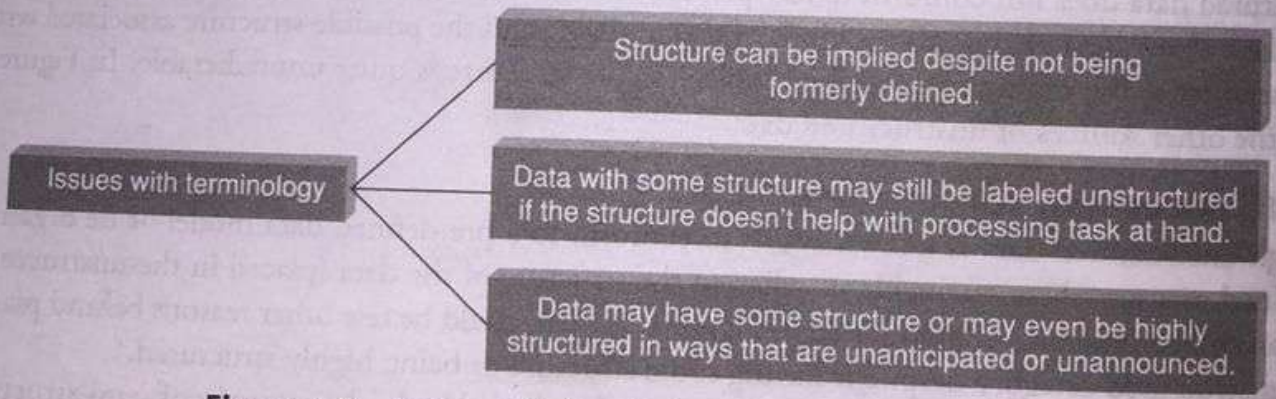

Figure 1.9 Issues with terminology of unstructured data.

one can easily look at the properties to get information such as the owner of the file, the date on which the file was created, the size of the file, etc. Okay, we do have little metadata. But when it comes to analysis, we are more concerned with the content of the text file rather than the name or any of the other properties. In fact, the other properties may not in any way contribute to the processing/analysis task at hand. Therefore, it is fair to place it in the unstructured data category.

1.1.3.2 How to Deal with Unstructured Data?

Today, unstructured data constitutes approximately 80% of the data that is being generated in any enterprise. The balance is clearly shifting in favor of unstructured data as shown in Figure 1.10. It is such a big percentage that it cannot be ignored. Figure 1.11 states a few ways of dealing with unstructured data.

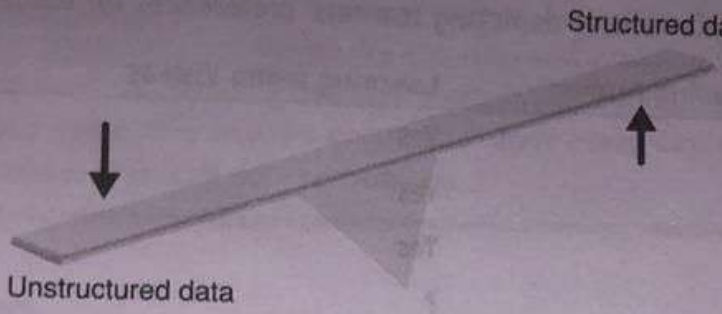

Figure 1.10 Unstructured data clearly constitutes a major percentage of enterprise data.

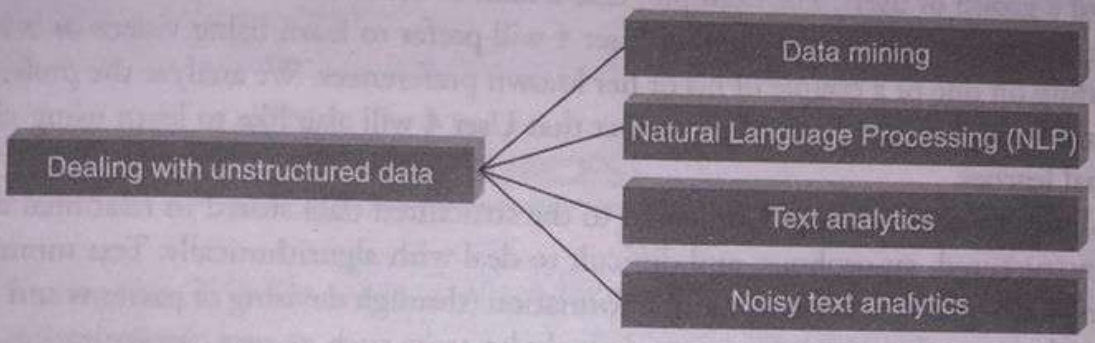

Figure 1.11 Dealing with unstructured data.

The following techniques are used to find patterns in or interpret unstructured data:

1. **Data mining:** First, we deal with large data sets. Second, we use methods at the intersection of artificial intelligence, machine learning, statistics, and database systems to unearth consistent patterns in large data sets and/or systematic relationships between variables. It is the analysis step of the "knowledge discovery in databases" process.

   Few popular data mining algorithms are as follows:

   - ***Association rule mining:*** It is also called "market basket analysis" or "affinity analysis". It is used to determine "What goes with what?" It is about when you buy a product, what is the other product that you are likely to purchase with it. For example, if you pick up bread from the grocery, are you likely to pick eggs or cheese to go with it.
   - ***Regression analysis:*** It helps to predict the relationship between two variables. The variable whose value needs to be predicted is called the dependent variable and the variables which are used to predict the value are referred to as the independent variables.

PICTURE THIS...

You are interested in purchasing real estate. You have been looking at a few good sites. You have come to the conclusion that cost of the real estate depends on the location (outskirts or prime locale), the amenities provided by the builder (joggers track, senior citizen zone, gymnasium, swimming pools, etc.), the built up area, etc. The cost of the real estate is the dependent variable and the location, amenities, built-up area are called the independent variables.

Table 1.5 Sample records depicting learners' preferences for modes of learning

	Learning using Audios	Learning using Videos	Textual Learners
User 1	Yes	Yes	No
User 2	Yes	Yes	Yes
User 3	Yes	Yes	No
User 4	Yes	?	?

- ***Collaborative filtering:*** It is about predicting a user's preference or preferences based on the preferences of a group of users. For example, take a look at Table 1.5.
  We are looking at predicting whether User 4 will prefer to learn using videos or is a textual learner depending on one or a couple of his or her known preferences. We analyze the preferences of similar user profiles and on the basis of it, predict that User 4 will also like to learn using videos and is not a textual learner.

2. **Text analytics or text mining:** Compared to the structured data stored in relational databases, text is largely unstructured, amorphous, and difficult to deal with algorithmically. Text mining is the process of gleaning high quality and meaningful information (through devising of patterns and trends by means of statistical pattern learning) from text. It includes tasks such as text categorization, text clustering, sentiment analysis, concept/entity extraction, etc.
3. **Natural language processing (NLP):** It is related to the area of human computer interaction. It is about enabling computers to understand human or natural language input.
4. **Noisy text analytics:** It is the process of extracting structured or semi-structured information from noisy unstructured data such as chats, blogs, wikis, emails, message-boards, text messages, etc. The noisy unstructured data usually comprises one or more of the following: Spelling mistakes, abbreviations, acronyms, non-standard words, missing punctuation, missing letter case, filler words such as "uh"; "um", etc.
5. **Manual tagging with metadata:** This is about tagging manually with adequate metadata to provide the requisite semantics to understand unstructured data.
6. **Part-of-speech tagging:** It is also called POS or POST or grammatical tagging. It is the process of reading text and tagging each word in the sentence as belonging to a particular part of speech such as "noun", "verb", "adjective", etc.
7. **Unstructured Information Management Architecture (UIMA):** It is an open source platform from IBM. It is used for real-time content analytics. It is about processing text and other unstructured data to find latent meaning and relevant relationship buried therein. Read up more on UIMA at the link: http://www.ibm.com/developerworks/data/downloads/uima/