Information retrieval systems are often contrasted with relational databases.

Traditionally, IR systems have retrieved information from unstructured text

– by which we mean “raw” text without markup. Databases are designed

for querying relational data: sets of records that have values for predefined

attributes such as employee number, title and salary. There are fundamental

differences between information retrieval and database systems in terms of

retrieval model, data structures and query language as shown in Table 10.1.

1

Some highly structured text search problems are most efficiently handled

by a relational database, for example, if the employee table contains an at-

tribute for short textual job descriptions and you want to find all employees

who are involved with invoicing. In this case, the SQL query:

select lastname from employees where job\_desc like ’invoic%’;

may be sufficient to satisfy your information need with high precision and

recall.

However, many structured data sources containing text are best modeled

as structured documents rather than relational data. We call the search over

STRUCTURED such structured documents structured retrieval. Queries in structured retrieval

RETRIEVAL can be either structured or unstructured, but we will assume in this chap-

ter that the collection consists only of structured documents. Applications

of structured retrieval include digital libraries, patent databases, blogs, text

in which entities like persons and locations have been tagged (in a process

called named entity tagging) and output from office suites like OpenOffice

that save documents as marked up text. In all of these applications, we want

to be able to run queries that combine textual criteria with structural criteria.

Examples of such queries are give me a full-length article on fast fourier transforms

(digital libraries), give me patents whose claims mention RSA public key encryption

1. In most modern database systems, one can enable full-text search for text columns. This

usually means that an inverted index is created and Boolean or vector space search enabled,

effectively combining core database with information retrieval technologies.