



Chapter-9

Boolean Query Model in practice
2021-2022

Retrieval Tasks

- Ad-hoc retrieval (querying) – pull technology
- Interactive query formulation
- Filtering – push technology
- Categorisation
- Clustering
- Search by browsing (hypertext)
- Search by browsing and querying (digital libraries and the web)

Document/Information/Knowledge

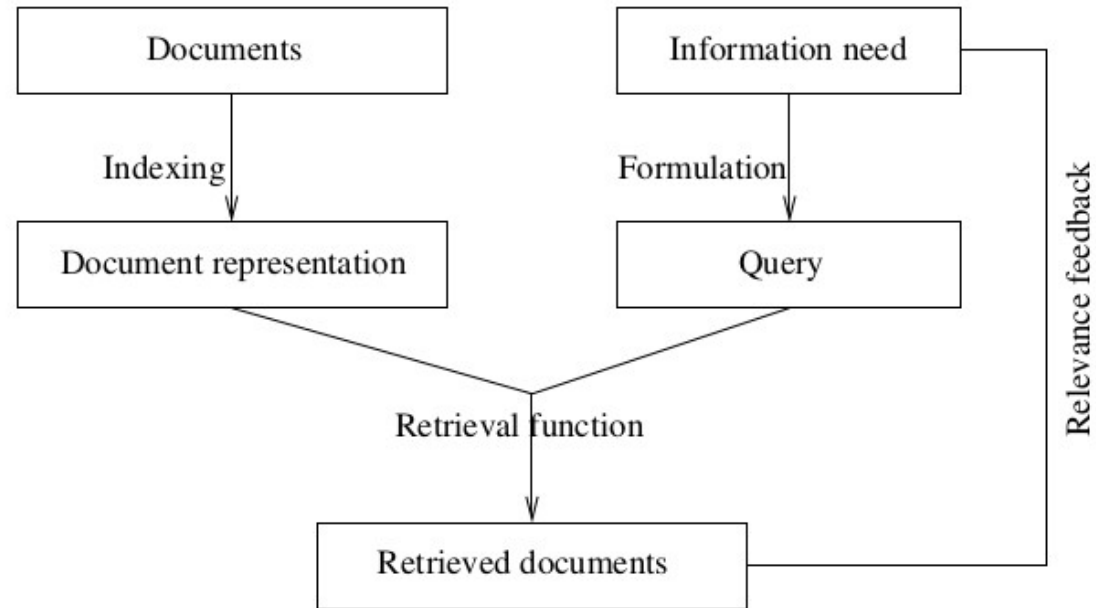
Retrieval/Management/Engineering

	document	information	knowledge
retrieval	indexing	ranking	reasoning
management	scanning	filtering	learning
engineering	structuring	modelling	annotating

Information Retrieval / Data Retrieval

	Information Retrieval	Data Retrieval
Matching	vague	exact
Model	probabilistic	deterministic
Query language	natural	artificial
Query specification	incomplete	complete
Items wanted	relevant	all (matching)
Error handling	insensitive	sensitive

A Conceptual Model for IR



Terminology and Symbols

D	A document or record.
D_i	Document i of a set of documents.
$\{D_i\}$	A set of documents, a file, or a database of documents.
$ D_i $	The cardinality or magnitude of the set $\{D_i\}$; the number of documents in the set.
$\{Dt_{i,k}\}$	A document is a set of terms, words, numbers, or character strings. The subscript k indicates a particular term in document i .
$ D_{i,k} $	The number of terms in document i .
Q	A query to an IRS for search of a database.
Q_j	Query j of a related sequence of queries.
$\{Q_j\}$	A series of queries all to accomplish one search objective.
$ Q_j $	The number of queries in the series.
$\{Qt_{j,l}\}$	A query also contains a set of terms. This designates term l of query j .
$ Qt_{j,l} $	The number of terms in query j .
S	The retrieval set, or drop, or “the set.” Those documents retrieved in response to a query, where which query is not specified.
S_j	The retrieval set for query j .
$ S_j $	The cardinality of set S_j ; the number of documents retrieved.
$S_{j,m}$	Document m in set S_j .
$St_{j,m,n}$	Term n of document m of the retrieval set for query j .

Boolean Retrieval

How to Build Inverted or (Indexed file)?

We will have a full example to illustrate how to use Boolean Retrieval model as to use Boolean query language steps.

An Example-step1

Doc-id	Data in Docs
1	To process solutions, the computer must have data. Data are unorganized facts. This information is printed in the form of reports.
2	The data the computer uses are of many different types. Computers must be told the data type of each variable or constant.
3	They go into the computer as input and are processed by the program. What is returned to the user is output, or information.

Step-2

No	block1	block2	block3	block4	block5
doc1	To process solutions	the computer must have data	Data are unorganized facts.	This information is printed	in the form of reports.

No	block1	block2	block3	block4	block5
doc2	The data the computer uses	are of many different types	Computers must be told	the data type of each	variable or constant

No	block1	block2	block3	block4	block5
doc3	They go into the computer	as input and are processed	by the program	What is returned to the user	is output or information

Step-3

doc1	process solution	compute data	Data organize fact	inform print	form report
------	------------------	--------------	-----------------------	--------------	-------------

doc2	data compute use	differ type	Compute told	data type	variable constant
------	------------------	-------------	--------------	-----------	----------------------

doc3	compute	input process	program	return use	output inform
------	---------	---------------	---------	------------	---------------

Step-4 Inverted file Initialization

Posting List table

t-id	term-frequency	doc-id
1	2	1,3
2	1	1
3	3	1,2,3
4	4	1,2
5
....
.....
.....		
.....		
13	3	2,3
30	2	1,3

No	term	doc-id
1	process	1
2	solution	1
3	compute	1
4	data	1
5	data	1
6	fact	1
7	organize	1
8	inform	1
9	print	1
10	form	1
11	report	2
12	data	2
13	compute	2
14	use	2
15	differ	2
16	type	2

17	compute	2
18	told	2
19	data	2
20	type	2
21	variable	2
22	constant	2
23	compute	3
24	input	3
25	process	3
26	program	3
27	return	3
28	use	3
29	output	3
30	inform	3

Boolean model

$$R(d, q) = \begin{cases} 1 & \text{if } d \rightarrow q \\ 0 & \text{otherwise} \end{cases}$$

Retrieve documents that make the query true.

Query (and document): logical combination of index terms

$q = (\text{computer} \wedge \text{data}) \vee (\text{information} \wedge \neg \text{process})$

computer = { d1, d2 }

Data = { d1, d2, d3 }

“Query evaluation” based on inverted file:

Information = { d1, d2, d3 }

Process = { d1 }

Negation “felt” to be useful – proved to be not effective! Leads to empty results!

No ranking: either a document is retrieved or not: {d 1 , d 2 , d 3 }

Boolean Model

Query q :

set of index terms and Document d : set of index terms

Ranking based on the cardinality of $d \cap q$, ie number of terms common to the document and the query

$$R(d, q) = |d \cap q|$$

For $q = \{ \text{computer, data, process} \}$, the outcome is a (partially ranked) list of document

$ d \cap q $	documents	rank
3	d1, d2	1
2	d3	3
1	d4	4



End of chapter-9

Any Question.....?