

Information Retrieval

- Overview

Contents

1. Data vs. Information Retrieval
2. Definitions
3. Concept
4. Requirements
 1. Query Effectiveness
 2. Query Efficiency
 3. Index Efficiency
 4. Scalability
5. Issues
 1. Models
 2. Searching & Ranking
 3. Query Processing

Data vs. Information Retrieval

Data Retrieval

- Precise description
- Well-structured data

- Precise results
- Yes-or-no results

Science !!

Information Retrieval

- Vague information need
- Natural Language, images,
...
- Semantic interpretation
- Approximate results
- Relevance ranking

Arts !!

Definitions

Collection : is a set of documents

Volume : is a subset of documents

Document : is a sequence of terms

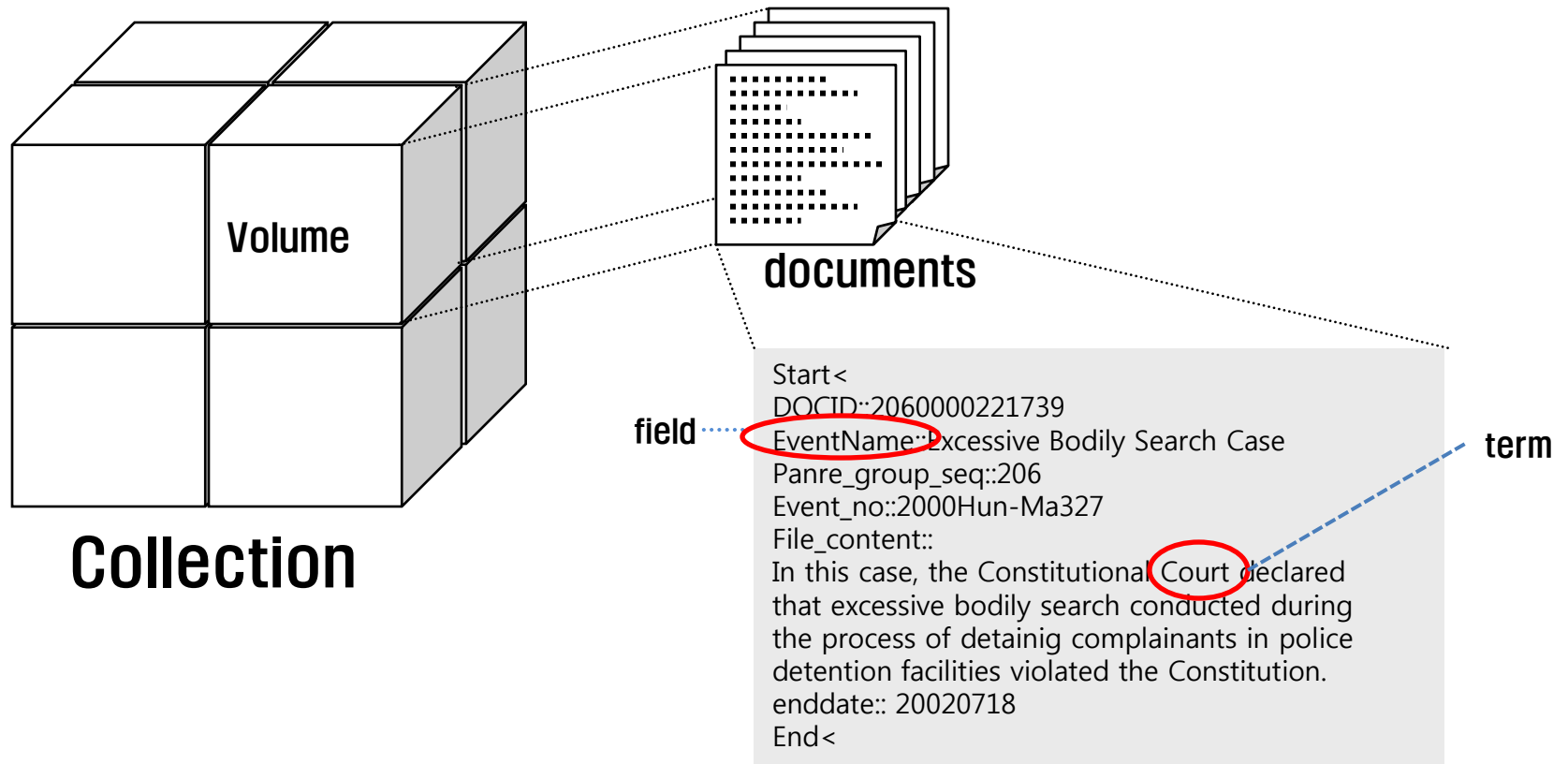
Term : is a semantic unit

ex) word, phrase, root of word

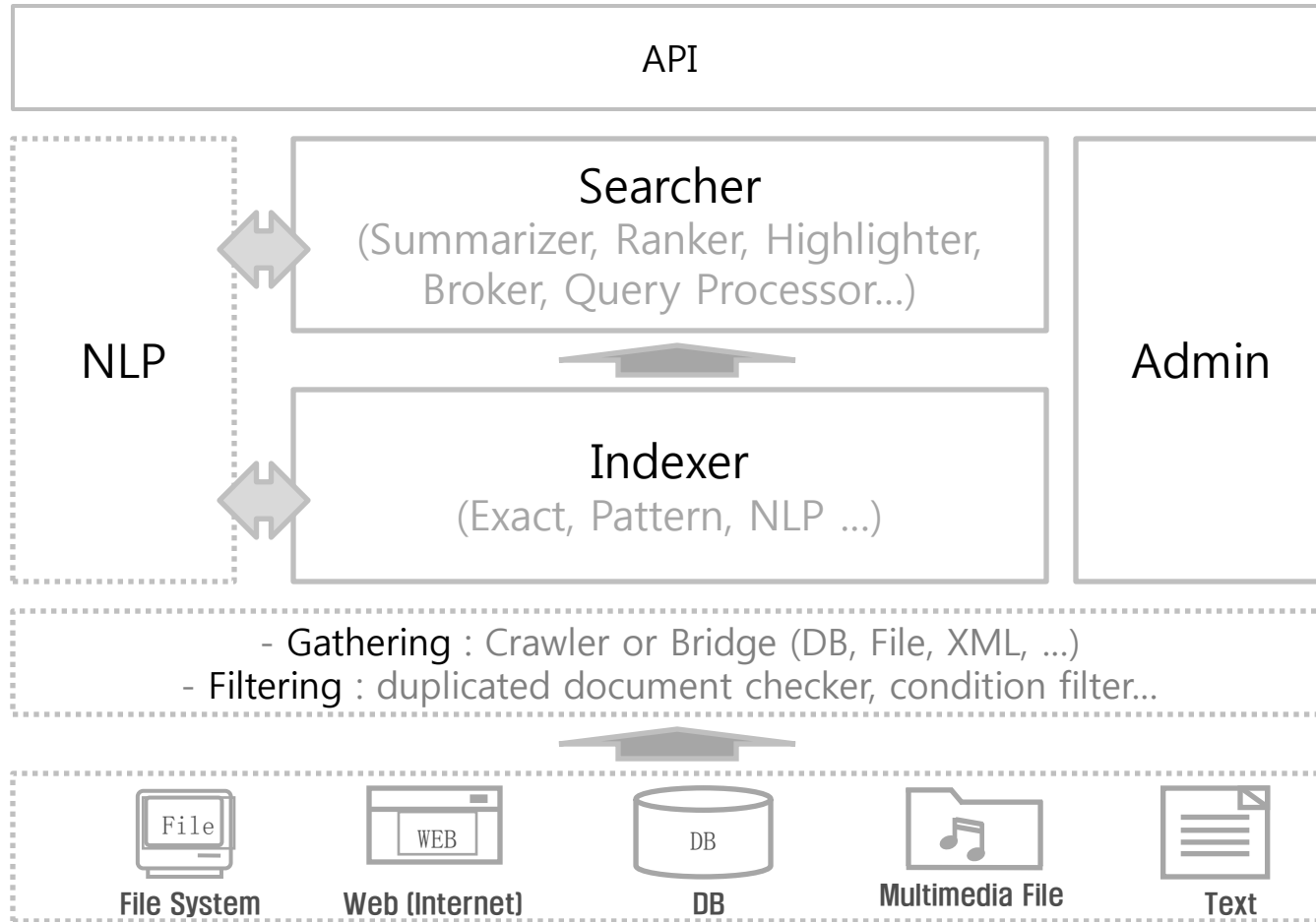
Query : is a request for documents pertaining to some topic

Information Retrieval (IR) System : attempt to find relevant documents to respond to a user's request

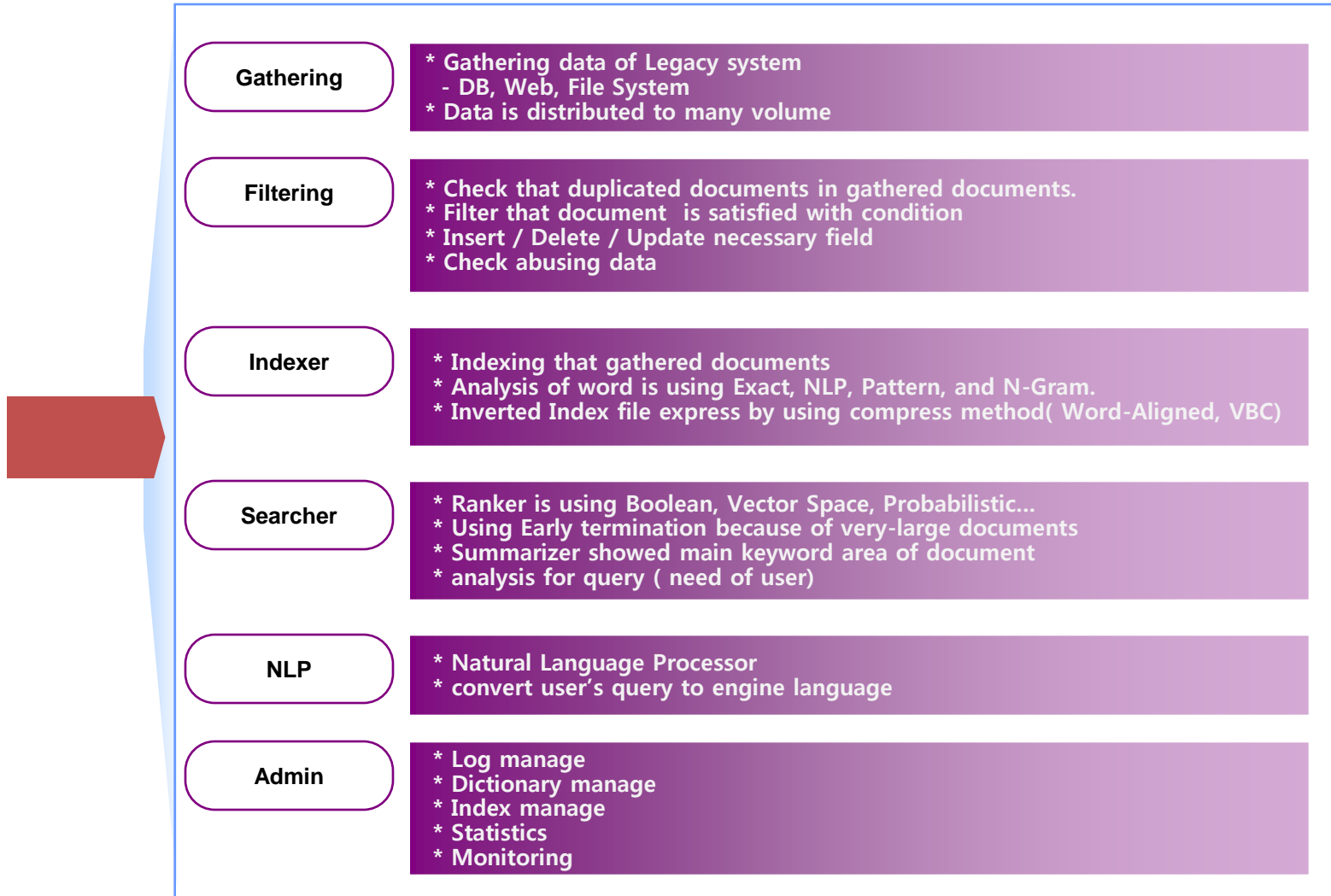
Data Class



Concept

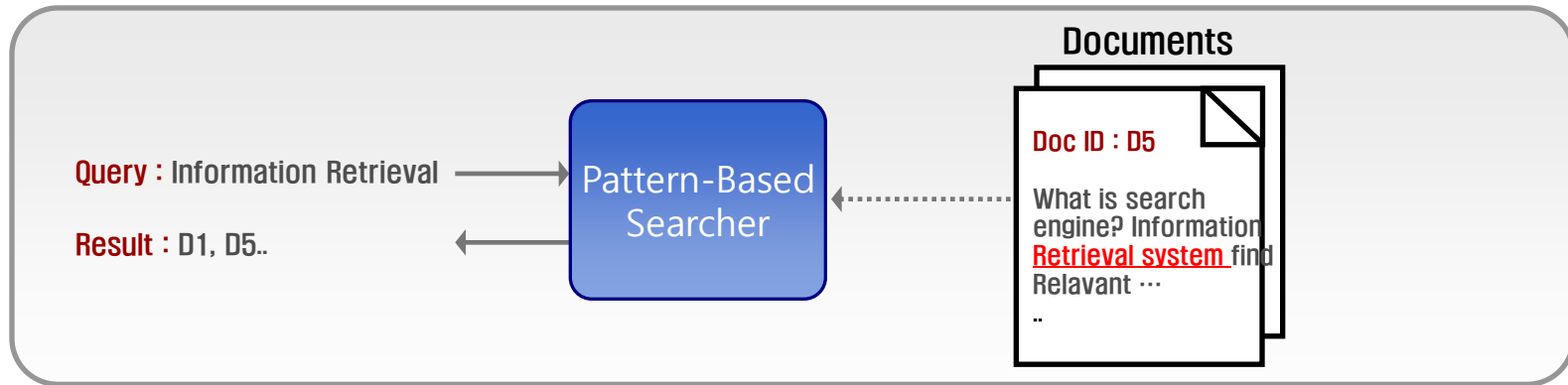


The role of each component

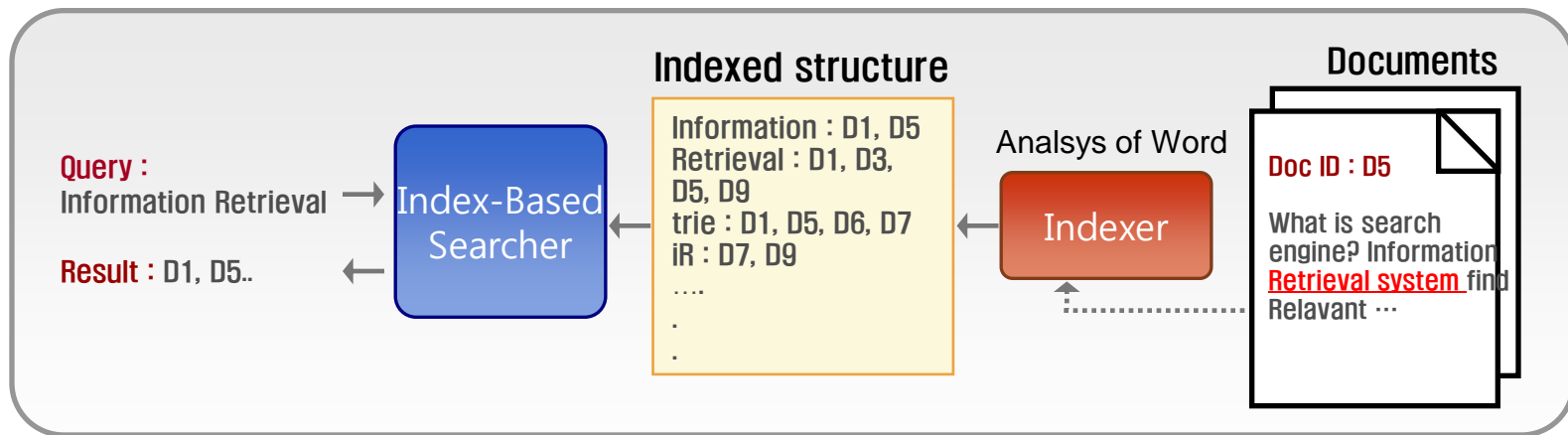


Basic Method for searching

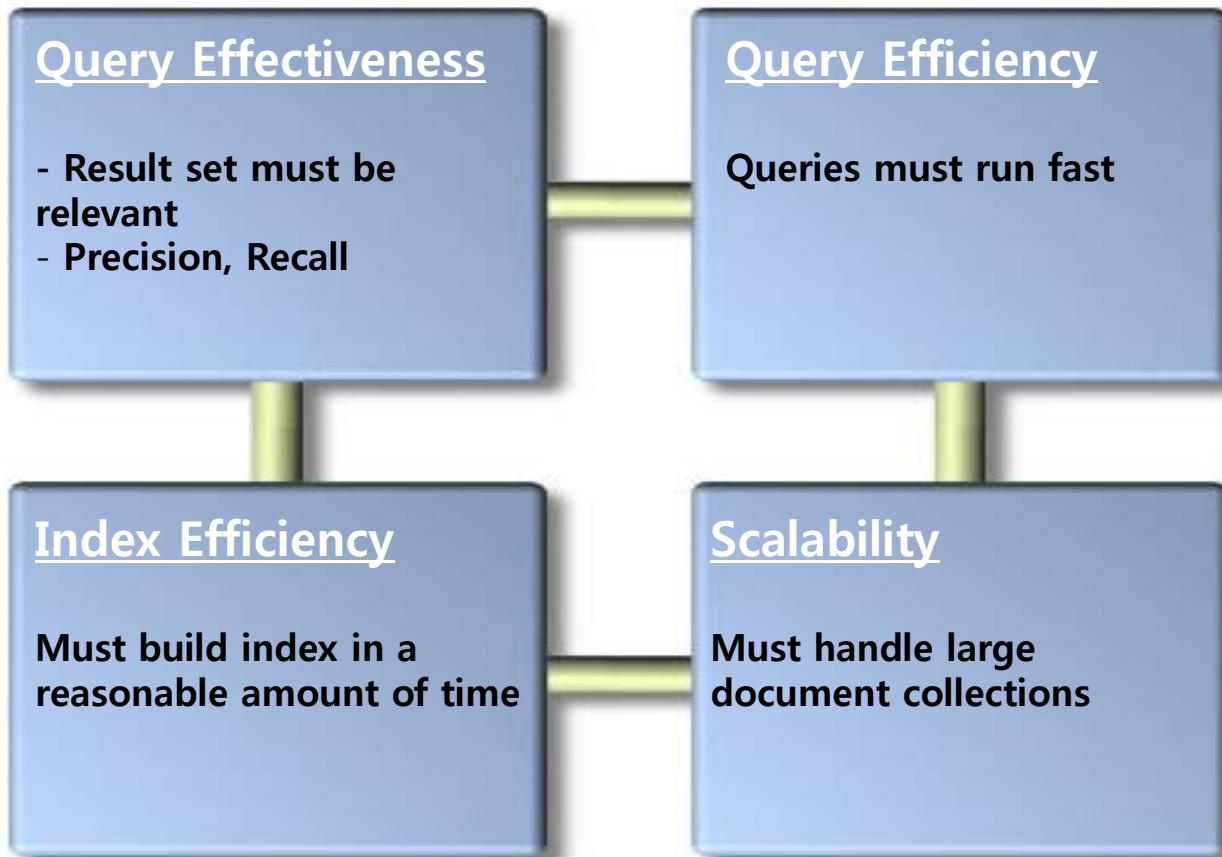
- **Patten-Based Matching Method** : searching sentences of document – small collection



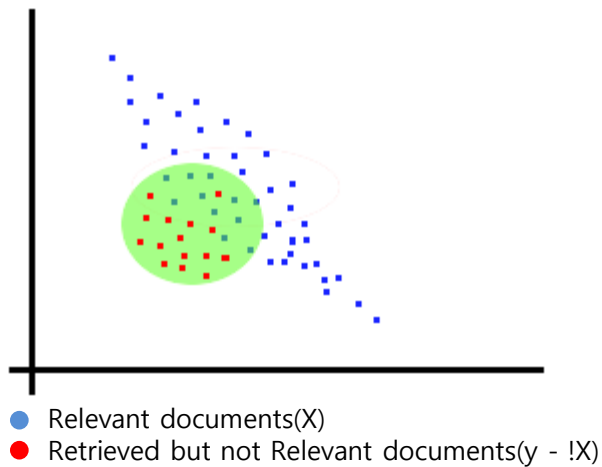
- **Index-Based Method** : inverted index structure



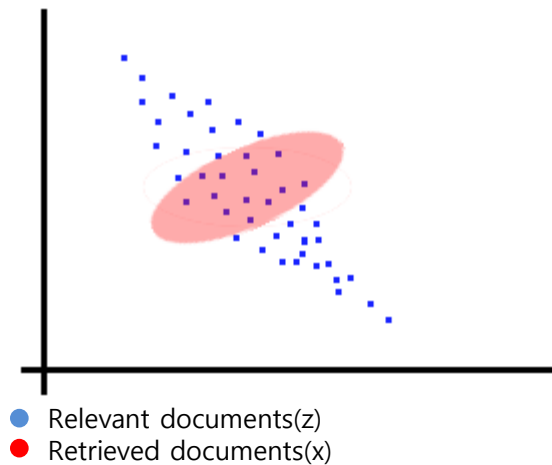
Requirements



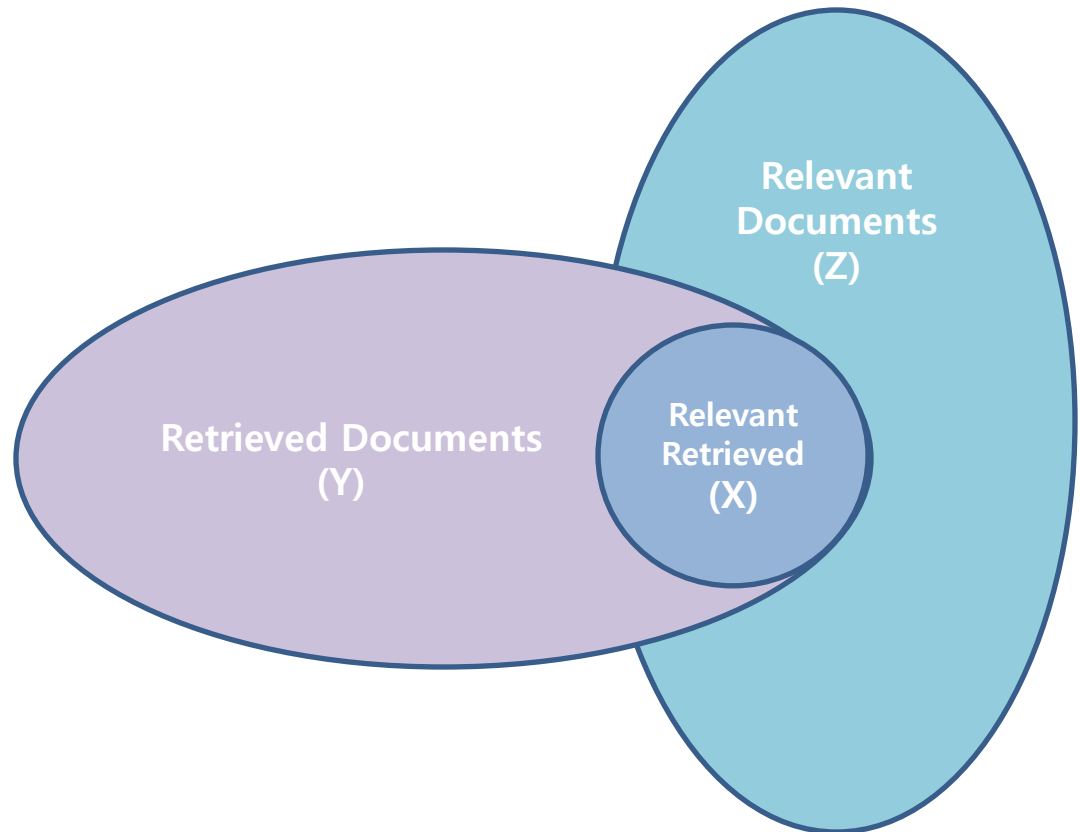
Query Effectiveness



Precision : x / y



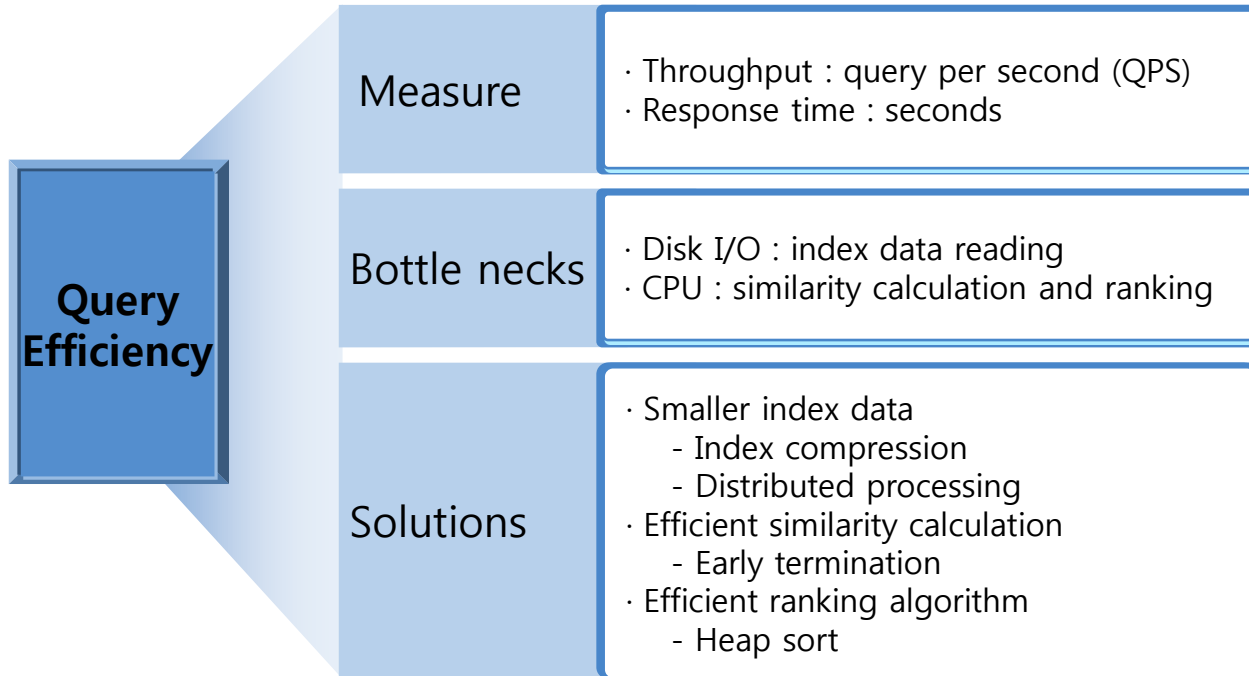
Recall : x / z



Relationship of Precision and Recall (generally)

- Recall \uparrow \rightarrow Precision \downarrow
showed relevant documents But, showed non-relevant documents also.
- Recall \downarrow \rightarrow Precision \uparrow
may not showed relevant documents

Query Efficiency



Index Efficiency

Issues on index data construction

- Finite memory
- Disk seek time

Solutions

- 1-pass strategy
 - Sort-based inversion
 - Merging
- 2-pass strategy
 - Preliminary pass
 - : Computing the number of terms, the number and size of index records for each term

Scalability

Very large document collections

- Google : about 20 billion pages
- Response slow??

Solution

- Parallel computing
 - MIMD (multiple instruction stream & multiple data stream)
- Multitasking vs. Partitioning
- Partitioning
 - Horizontal(document) partitioning
 - union of results
 - Vertical(term) partitioning
 - intersect of results

Models

Boolean Model	Vector Space Model	Probabilistic Model
<ul style="list-style-type: none">- Set theoretic- extended	<ul style="list-style-type: none">- Algebraic- Generalized vector- Latent semantic indexing (LSI)- Neutral network	<ul style="list-style-type: none">- Inference network- Belief network

Strength and weakness of each Model

Boolean

-Operations
AND, OR, NOT~AND

- Clear semantics
- Neat formalism
- Simple

- No ranking
- Retrieves too many
or too few
- No term weighting

- Extended
Term weighting

Vector Space

- Documents and queries are
mapped into term vector space
-Weighting
TF, DF, IDF
- Normalized by considering
document length

- Quality(term weighting)
- approximate matching(partial
matching)
- Ranking(similarity measures : inner
product, cosine coefficient, etc.)
-Simple and fast

- No logical expressions
- No term dependencies
- Large documents are somewhat
penalized

-Latent semantic indexing
* index by larger units : concepts
algebraic combination of set of
terms used together
* eliminates unimportant details
* lower dimensional space

Probabilistic

- Assumptions
* set of priori relevant
documents
* probabilities of documents
to be relevant
* After Bayes calculation :
probabilities of terms to
be important for defining
relevant documents
- Alternative to priori relevant
information
* Relying on user's feedback
* Without any relevance
information

- Theoretical basis

-Piori information
- Binary weight, ignore
frequencies
modification :
incorporating it-idf and
document length

Searching and Ranking



Retrieve matched documents on inverted indexes

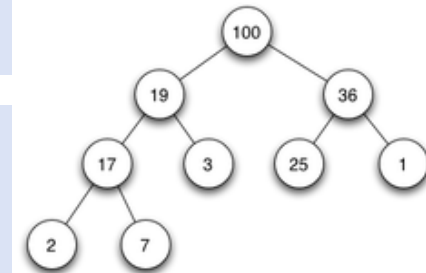
- Single term : locate entry, retrieve documents from list
- Conjunction of terms : intersection of lists
- Disjunction of terms : union of lists
- Negation of terms : complement of lists

Types of Ranking

- Relevance between a query and a document
- Features of documents : date, size, importance of a document, etc.

Ranking with heap

- Heap
 - A specialized tree-based data structure
 - key of a node is less than key of its parent
 - Time complexity : $O(n \log n)$
- Ranking
 - A heap maintains only top m-documents with reversed order
 - Each documents is compared to the document in root node.
 - Replace root node and heapify when new document is bigger.
- Benefit of using heap in ranking
 - Time complexity to get top m-documents : $O(n \log m)$



Query Processing

Wildcard Query

- mon* : find all documents containing any word beginning "mon"
- Union of terms : retrieve all word in range (mon \leq w < moo)
- *mon : nearly impossible
- Permuterm index
 - * for word hello index under :
 - * hello\$, ello\$h, llo\$he, lo\$hel, o\$hell
 - * queries

* <u>X</u> : lookup on X\$,	<u>X*</u> : lookup on \$X*
* * <u>X</u> : lookup on X\$*	* <u>X*</u> : lookup on X*
* <u>X*Y</u> : lookup on Y\$X*	
- n-gram index
 - * for word hello index under (bigram index) :
 - * \$h, he, el, ll, lo, o\$
 - * queries

* <u>hel</u> : \$h {AND} he {AND} el {AND} I\$,	<u>hel*</u> : \$h {AND} he {AND} el
* * <u>hel</u> : he {AND} el {AND} I\$	* <u>hel</u> : he {AND} el
* <u>he*lo</u> : \$h {AND} he {AND} lo {AND} o\$	

Query Analysis

- detect user intension from query and expand search request
- ex1) "dentist's near KangNam stop"
 - * collection : local information
 - * keyword : dentist
 - * sorting : distance from KangNam stop \rightarrow point of interest
- ex2) "latest song of The Beatles"
 - * collection : music
 - * keyword : Beatles
 - * target field : singer
 - * sorting : descend of date