Chap # 01

Boolean Retrieval. ①

Date _____

* Information retrieval (IR) is defined as ;
finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections
Adhoc Retrieval task, most standard IR task.
* Example : Which plays of shakespeare contain the words Brutus & Ceasar but not Calpurnia?

Grep all shakespeare's play for 'Brutus' & 'Ceasar', then strip out lines containing 'Calpurnia'.

* Boolean Retrieval model :=
It is a model for information retrieval in which we can pose any query which is in the form of boolean expression of terms
Model views each document as just a set of words.

* Term - Document Incidence Matrix :-
It is a matrix of size |t| x |d|.
Matrix element (t, d) is 1 if play in column d contains the word in row t, otherwise 0.

To answer the queries we take vectors and perform boolean operation.

nice

Date

Problems:

① Sparse matrix : Very large matrix with few non-zero entries.

② Exact matching : Difficult to match similar surface features.

eg: operate, operational, operation etc.

③ Complex query information

Solution : Inverted Index

* Adhoc Retrieval System:

Goal: To provide documents from within the collection that are relevant to an arbitrary user information need, communicated to the system by-means of a one-off, user initiated query. Example: Internet search engine

* Effectiveness of IR System

The quality of IR's search system.

Two factors are;

① Precision: (Ratio of relevant from result)

What fraction of the returned results are relevant to the information need?

$$= \frac{\text{Relevant - retrieved}}{\text{Total retrieved}} \qquad \frac{TP}{TP+FP}$$

Date

→ Building an inverted index

① Initially we have a list of pairs of term and docIDs.
② We sort this list, so that the terms are in alphabetical order.
③ Multiple occurences of same term from the same document are then merged.
④ Instances of same term are then grouped, and the result is split into dictionary & posting. Dictionary also contains document frequency.

Dictionary is kept on the memory
Posting list is kept on the disk

Q What data structure should we use for a posting list ?

① Fixed length array → waste memory
② Singly linked list
' Cheap insertions
For advanced strategies like skip list, it requires additional pointers
③ Variable length array
Avoid overhead for pointers
Contiguous memory increases speed

nice

Date

* Processing Boolean Queries:

Q How do we process a query using an inverted index and the basic boolean retrieval model?

Consider the Simple Conjunctive Query:

① Brutus & Calpurnia

→ locate Brutus in dictionary
→ Retrieve its postings P1
→ locate Calpurnia in dictionary
→ Retrieve its postings P2
→ Intersect the two posting lists.

```
Intersection(P1, P2)                    O(x+y) operations.
{                                       x = length of posting P1
    answer ← ( )                        y = length of posting P2
    While P1 ≠ NULL and P2 ≠ NULL
do if docID(P1) = = docID(P2)           Querying => O(N)
        then ADD(answer, docID(P1))     N = no. of doc in collection
             P1 ← next(P1)
             P2 ← next(P2)
    else if docID(P1) < docID(P2)
             then P1 ← next(P1)
             else P2 ← next(P2)
    return answer;
}
```

nice

Date

② Brutus AND Ceasar AND Calpurnia

Query Optimization:

→ Process of selecting how to organize the work of answering a query so that the least amount of work needs to be done by the system.

→ Aim is to select best order for query processing to minimize the amount of work

→ Best order is to process terms in order of increasing document frequency

Brutus = 2, Ceasar = 3, Calpurnia = 4

(Brutus ∧ Ceaser) ∧ Calpurnia

2 3 4

2

2

③ (madding OR Crowd) AND (ignoble OR strife) AND (killed OR slain).

We have to evaluate & temporarily store the answers for intermediate expressions in a complex expression.

Conjunctive normal form ← $(t_1 \vee t_2) \wedge (t_3 \vee t_4)$
Disjunctive normal form ← $(t_1 \wedge t_3) \vee (t_1 \wedge t_4) \vee (t_2 \wedge t_3) \vee (t_2 \wedge t_4)$

Date

In this case, it is more efficient to intersect each retrieved postings list with the current intermediate results in the memory, where we initialize the intermediate result by loading the posting list of the least-frequent term

```
Optimized Intersection (<t1, ..., tn>)
{
    terms ← SortByIncreasingFrequency (<t1 ... tn>)
    result ← postings (first(term))
    terms ← rest(terms)
    while terms ≠ NULL and result ≠ NULL
    do result ← Intersect (result, postings (first(term)))
        terms ← rest (terms)
    return result
}
```

* Boolean Model

Assumptions:
→ User knows what type of document is used
→ Documents contains features (words, phrases etc)
→ User's query is about these features

Date

Advantages:
Clean, formal, easy-to implement
Predictable results
Incorporate many features.

Disadvantages:
Flat results - No ranking
Equally weighted terms
Exact matching - Retrieves too few or too many results.

Observations:
- No partial matching
- Binary criterion for deciding relevance
- Flat results
- Information need has to be translated into Boolean expression.
- Returns too few or too many documents in response.
- Boolean queries formulated by users are most often too simplistic.

* Extended Boolean Model.
- Assigns weight to every term
- Covers association between terms
- Flat results, no ranking

* Fuzzy set model
- Assigns fuzzy score.

nice

Date

$\forall_{d_i=0}^{m} (\vec{d} + \vec{q} = \vec{q}) \Rightarrow (Result = d_k)$

$d_k = <1 0 1\ 0\ 0\ 1 1>$ → no ranking

$d_k = <2 0 3\ 0\ 0\ 1\ 4>$ → no. of occurences
$T_1\ T_2\ T_3\ T_3\ T_4\ T_5\ T_6$

(Food for thoughts) Chap # 01

① Difference between text mining & data mining

Text Mining	Data Mining
Extracting information contained in a textual document	Extracting information from a dataset.
Textual data	Numerical/Categorical data
Semi structured/ Unstructured	Structured
IR, NLP, data mining	Data analysis, neural networks

Date _____

② Differentiate among Data, Information, knowledge and Wisdom?

Data ← Raw facts and figures
Information ← Processed form of data
Knowledge ← We know a subject or; we know where we can find info upon it. Appropriate collection of info.
Wisdom ← Evaluated understanding.

③ Information retrieval & its components.

Answer # pg 01

Components of IR system

① Query / Collection
Store only a representation of the document or query

② IR system
Involve in performing actual retrieval function, executing search strategy in response to a query.

③ Ranked Result
Set of documents which improves the subsequent run after IR.

④ and ⑤ Answer # pg 2, 3

⑥ Answer # pg # 2.

Date_____

(7) What are the best features of West-Law Commercial Boolean Retrieval System and what are some of its drawbacks

Idk

- Precise queries
- Use of proximity operators
- Queries average about 10 words in length.
- Space b/w words represent disjunction
- Boolean queries are precise, either matches a document or not
- Allows an effective means of document ranking within a boolean model.

→ 'And' ya 'intersection' wali query.

(8) For a conjunctive query, is processing postings lists in order of size guaranteed to be optimal?

Processing posting list in order of size is good approach but Not guaranteed to be optimal.

Counter example :=

t_1 - term1 → 1, 2, 3

t_2 - term2 → 2, 3, 4, 5

t_3 - term 3 → 10, 11, 20, 30, 50

→ yahan ziada comparision krna parh rha hai.

$(t_1 \wedge t_2) \wedge t_3$ → requires more steps

$(t_1 \wedge t_3) \wedge t_2$ → requires less steps b/c there is in no intersection between 1st & 3rd postings list.

nice

Date

② Recall: (Ratio of relevant from expected)
What fraction of relevant documents in the collection were returned by the system?

$$= \frac{\text{relevant - retrieved}}{\text{total relevant in doc}} \qquad \frac{TP}{TP + FN}$$

→ 100% precision
Atleast 1 relevant from collection $\frac{1}{1} \times 100$.

→ 100% recall
send all relevant. $\frac{1}{1} \times 100$

		Actual: Good	Actual: Bad		Rel	Non Rel
Predicted	Good	TP	FP	Ret	TP	FP
	Bad	FN	TN	Non Ret	FN	TN

* Inverted Index
We have a dictionary of terms and for each term, we have a list of documents in which the term t occurs.

Dictionary of words → Vocabulary / lexicon
Document information → Posting list / Posting.