Information Retrieval
Tutorial 1: Boolean Retrieval

Professor: Michel Schellekens
TA: Ang Gao

University College Cork

2012-10-26

Outline

1. Review
2. Exercises

Definition of *information retrieval*

What is IR ?

Definition of *information retrieval*

What is IR ?

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Definition of *information retrieval*

What is IR ?
Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Definition of *information retrieval*

What is IR ?
Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Definition of *information retrieval*

What is IR ?
Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Definition of *information retrieval*

What is IR ?

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Definition of *information retrieval*

What is IR ?
Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Effectiveness of an IR system

Effectiveness of an IR system

- Precision : Fraction of retrieved docs that are relevant to user's information need

Effectiveness of an IR system

- Precision : Fraction of retrieved docs that are relevant to user's information need
- Recall : Fraction of relevant docs in collection that are retrieved

Boolean retrieval

- The Boolean model is arguably the simplest model to base an information retrieval system on.

Boolean retrieval

- The Boolean model is arguably the simplest model to base an information retrieval system on.
- Queries are Boolean expressions, e.g., CAESAR AND BRUTUS

Boolean retrieval

- The Boolean model is arguably the simplest model to base an information retrieval system on.
- Queries are Boolean expressions, e.g., CAESAR AND BRUTUS
- The seach engine returns all documents that satisfy the Boolean expression.

Term-document incidence matrix

To build IR system we need index the documents in advance.

Term-document incidence matrix

To build IR system we need index the documents in advance.

Term-document incidence matrix

- Terms are the indexed units(usual words).

Term-document incidence matrix

To build IR system we need index the documents in advance.

Term-document incidence matrix

- Terms are the indexed units(usual words).
- Column: a vector for each document, showing the terms that occur in it.

Term-document incidence matrix

To build IR system we need index the documents in advance.

Term-document incidence matrix

- Terms are the indexed units(usual words).
- Column: a vector for each document, showing the terms that occur in it.
- Row: a vector for each term, which shows the documents it appears in.

Term-document incidence matrix

To build IR system we need index the documents in advance.
Term-document incidence matrix

- Terms are the indexed units(usual words).
- Column: a vector for each document, showing the terms that occur in it.
- Row: a vector for each term, which shows the documents it appears in.
- Query: Answer boolean expression of terms, do bitwise AND OR and NOT on vectors eg: 110100 and 110111 and 101111 $=$ 100100.

Term-document incidence matrix

To build IR system we need index the documents in advance.

Term-document incidence matrix

- Terms are the indexed units(usual words).
- Column: a vector for each document, showing the terms that occur in it.
- Row: a vector for each term, which shows the documents it appears in.
- Query: Answer boolean expression of terms, do bitwise AND OR and NOT on vectors eg: 110100 and 110111 and 101111 $=$ 100100.

Term-document incidence matrix

To build IR system we need index the documents in advance.

Term-document incidence matrix

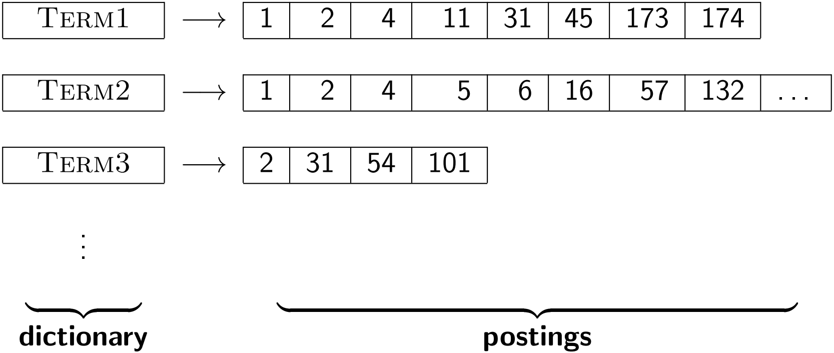
- Terms are the indexed units(usual words).
- Column: a vector for each document, showing the terms that occur in it.
- Row: a vector for each term, which shows the documents it appears in.
- Query: Answer boolean expression of terms, do bitwise AND OR and NOT on vectors eg: 110100 and 110111 and 101111 = 100100.

	Doc1	Doc2	Doc3	Doc4	Doc5	...
TERM1	1	1	0	0	0	
TERM2	1	1	0	1	0	
TERM3	1	1	0	1	1	
TERM4	0	1	0	0	0	
TERM5	1	0	0	0	0	
...						

Entry is 1 if term occurs.

Inverted Index

For each term t, we store a list of all documents that contain t.



TERM1 ⟶ 1 | 2 | 4 | 11 | 31 | 45 | 173 | 174

TERM2 ⟶ 1 | 2 | 4 | 5 | 6 | 16 | 57 | 132 | ...

TERM3 ⟶ 2 | 31 | 54 | 101

⋮

dictionary

postings

Inverted index construction

1. Collect the documents to be indexed:
   Friends, Romans, countrymen. So let it be with Caesar . . .
2. Tokenize the text, turning each document into a list of tokens:
   Friends Romans countrymen So . . .
3. Do linguistic preprocessing, producing a list of normalized tokens, which are the indexing terms: friend roman
   countryman so . . .
4. Index the documents that each term occurs in by creating an inverted index, consisting of a dictionary and postings.

Intersecting two postings lists

$$\textsc{Term1} \longrightarrow \boxed{1} \rightarrow \boxed{2} \rightarrow \boxed{4} \rightarrow \boxed{11} \rightarrow \boxed{31} \rightarrow \boxed{45} \rightarrow \boxed{173} \rightarrow \boxed{174}$$

$$\textsc{Term2} \longrightarrow \boxed{2} \rightarrow \boxed{31} \rightarrow \boxed{54} \rightarrow \boxed{101}$$

$$\text{Intersection} \Longrightarrow \boxed{2} \rightarrow \boxed{31}$$

- This is linear in the length of the postings lists.
- Note: This only works if postings lists are sorted.

Intersecting two postings lists

```
INTERSECT($p_1, p_2$)
answer $\leftarrow \langle\ \rangle$
while $p_1 \neq$ NIL and $p_2 \neq$ NIL
do if $docID(p_1) = docID(p_2)$
      then ADD($answer, docID(p_1)$)
           $p_1 \leftarrow next(p_1)$
           $p_2 \leftarrow next(p_2)$
      else if $docID(p_1) < docID(p_2)$
             then $p_1 \leftarrow next(p_1)$
             else $p_2 \leftarrow next(p_2)$
return answer
```

Outline

1 Review

2 Exercises

Question1

Consider these documents:

Doc1 breakthrough drug for schizophrenia

Doc2 new schizophrenia drug

Doc3 new approach for treatment of schizophrenia

Doc4 new hopes for schizophrenia patients

- draw the term-document incidence matrix for this document collection
- draw the inverted index representation for this collection.
- what are the returned results for these queries:
  - schizophrenia AND drug
  - for AND NOT(drug OR approach)

Solution:1.a

	Doc1	Doc2	Doc3	Doc4
approach	0	0	1	0
breakthrough	1	0	0	0
drug	1	1	0	0
for	1	0	1	1
hopes	0	0	0	1
new	0	1	1	1
of	0	0	1	0
patients	0	0	0	1
schizophrenia	1	1	1	1
treatment	0	0	1	0

Solution:1.b

approach $\longrightarrow \boxed{3}$

breakthrough $\longrightarrow \boxed{1}$

drug $\longrightarrow \boxed{1} \rightarrow \boxed{2}$

for $\longrightarrow \boxed{1} \rightarrow \boxed{3} \rightarrow \boxed{4}$

hopes $\longrightarrow \boxed{4}$

new $\longrightarrow \boxed{2} \rightarrow \boxed{3} \rightarrow \boxed{4}$

of $\longrightarrow \boxed{3}$

patients $\longrightarrow \boxed{4}$

schizophrenia $\longrightarrow \boxed{1} \rightarrow \boxed{2} \rightarrow \boxed{3} \rightarrow \boxed{4}$

treatment $\longrightarrow \boxed{3}$

Solution:1.c

schizophrenia $\longrightarrow$ $\boxed{1} \rightarrow \boxed{2} \rightarrow \boxed{3} \rightarrow \boxed{4}$

drug $\longrightarrow$ $\boxed{1} \rightarrow \boxed{2}$

AND $\longrightarrow$ $\boxed{1} \rightarrow \boxed{2}$

Solution:1.c

for $\longrightarrow$ $\boxed{1} \rightarrow \boxed{3} \rightarrow \boxed{4}$

approach $\longrightarrow$ $\boxed{3}$

drug $\longrightarrow$ $\boxed{1} \rightarrow \boxed{2}$

for AND NOT(drug OR approach) $\longrightarrow$ $\boxed{4}$

Question 2

Recommend a query processing order for

(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)

given the following postings list sizes:

Term	**Postings size**
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

Solution 2

First we approximate the OR operator with the sum of the frequencies and then execute the query from lowest frequency to highest.
(kaleidoscope OR eyes) (300,321) AND (tangerine OR trees) (363,465) AND (marmalade OR skies) (379,571)

Question 3

Write out a postings merge algorithm for an x OR y query

Solution 3

```
$\text{UNION}(p_1, p_2)$
    $answer \leftarrow \langle\ \rangle$
    while $p_1 \neq \text{NIL}$ and $p_2 \neq \text{NIL}$
    do if $docID(p_1) = docID(p_2)$
          then $\text{ADD}(answer, docID(p_1))$
               $p_1 \leftarrow next(p_1) p_2 \leftarrow next(p_2)$
          else if $docID(p_1) < docID(p_2)$
                  then $\text{ADD}(answer, docID(p_1))$
                       $p_1 \leftarrow next(p_1)$
                  else $\text{ADD}(answer, docID(p_2))$
                       $p_2 \leftarrow next(p_2)$
    while $p_1 \neq \text{NIL}$
    do $\text{ADD}(answer, docID(p_1))$
       $p_1 \leftarrow next(p_1)$
    while $p_2 \neq \text{NIL}$
    do $\text{ADD}(answer, docID(p_2))$
       $p_2 \leftarrow next(p_2)$
    return $answer$
```