AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

National University of Computer & Emerging Sciences
FAST-Karachi Campus
Information Retrieval (CS317)
Quiz#1

Dated: February 12, 2020 Marks: 20
Time: 20 min.
Std-ID: ______Sol_______

Question No. 1

What are some of the assumptions we made while developing Boolean Model for IR? Outline them. [5]

The model assume that users know the features from the document.

⇨ Obvious this is not true, as users may not aware of all the lexical features present in a document. The query term may contain some synonyms and for this retrieved result may surprise the users.

The documents are available in machine readable format.

⇨ documents are available in machine readable format is far from simplifications some human languages pose many challenges in processing the digital documents for indexing the Boolean features.

Question No.2

There are 16 relevant documents in a collection for a given query "q". The precision of the query is 0.40 and recall of the query is 0.25, Find, how many documents are in the results-set (number of documents response to the query "q" from the IR system)? [5]

we know,
precision = (relevant-retrieved) / (total-retrieved)
=> 0.4 = (relevant-retrieved) / (result-set)
=> (result-set) = (relevant-retrieved) /0.4 ---------- eq(A)
similarly,
recall = (relevant-retrieved)/ (total-relevant)
=> 0.25 = (relevant-retrieved)/ 16
=> relevant-retrieved= 0.25 * 16 = 4
hence
eq(A) => result-set = 4 /0. 4 = 10
Result-set has 10 documents.

Question No.3

Comments on the following statement as TRUE or FALSE with justification (1-2 line explanations). 5 X 2 marks each.

1. **Stemming decreases, the size of the lexicon(dictionary).**

   TRUE. Stemming maps derivational /inflectional morphemes to root word suing heuristic based algorithms hence decreases size of the dictionary.

2. **A general phrase query cannot be answered by bi-words index.**

   TRUE. A general phrase query is of the form "w1 w2 …wn". A bi-words index is of the form that index terms like w1w2, w2w3, etc. It can fetch documents that contains these bi-words but not necessary in the given sequences or (in given string order).

3. **Extended bi-words index is not very useful.**

   FALSE. For some of the languages like English, Morphological or syn-set of words are such that some words are more commonly appears together in the text. Keeping a bi-index on these bi-words is beneficial for IR. Example: Coins are in my pocket => coin pocket

4. **Positional Index is good for proximity query.**

   TRUE. A proximity query is of the form "labor policy /k". The intent of the user is to get the documents that contains both the words "labor" and "policy" within k words apart in the documents. Positional Index can be used to answer this type of query.

5. **Text is always a linear sequence of characters in electronic form.**

   TRUE. All languages are based on some character set, these characters are concatenated to form words and words are joined together to form sentences. These are all long runs of binary digits in the core form. Hence linear in nature.

BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

National University of Computer & Emerging Sciences
FAST-Karachi Campus
Information Retrieval (CS317)
Quiz#1

Dated: February 12, 2020 Marks: 20
Time: 20 min.
Std-ID: ______Sol________

Question No. 1

What are some of the drawbacks of Boolean Model for IR? With respect to users and systems perspectives. [5]

There are several drawbacks of Boolean Model for IR.

From Users prospective:

Users need to educate on query formulations, they need to understand Boolean queries. They need to have some clear idea about what features are there in the relevant documents.

From System's prospective:

The IR systems based on Boolean model considers all terms with same importance. It is based on exact matching and result-set is flat (that is all documents are equally ranked).

Question No.2

There are 32 relevant documents in a collection for a given query "q". The recall of the query is 0.25 and precision of the query is 0.40, Find, how many documents are in the results-set (number of documents response to the query "q" from the IR system)? [5]

we know,
precision = (relevant-retrieved) / (total-retrieved)
=> 0.4 = (relevant-retrieved) / (result-set)
=> (result-set) = (relevant-retrieved) /0.4 ---------- eq(A)
similarly,
recall = (relevant-retrieved)/ (total-relevant)
=> 0.25 = (relevant-retrieved)/ 32
=> relevant-retrieved= 0.25 * 32 = 8
hence
eq(A) => result-set = 8 /0. 4 = 20
Result-set has 20 documents.

Question No.3

Give an example of a query(text) for each type along with the best data structures to process these query efficiently with an inverted index. [2 x 5]

a. **General phrase query**

Consider the query “Cross Language Information retrieval” it is a general phrase query. The intent of the user is to get the documents that contains the complete list of words in the same order. Positional Index can be used to answer this type of query.

b. **Proximity query**

Consider the query “labor policy /k” it is proximity query. The intent of the user is to get the documents that contains both the words “labor” and “policy” within k words apart in the documents. Positional Index can be used to answer this type of query.

c. **Trailing wildcard query**

Consider the query “mon*” it is an example of trailing wildcard query. The intent of the user is to get the documents that contains the words that has prefix of (mon) in it. The most suitable data structure for such queries are B-Tree or B+-Tree as it can offer quick access to all such terms.

d. **General wildcard query**

Consider the query “re*o*ting” it is a general wild card query. The intent of the user is to get the documents that contains the words containing the prefix “re” and suffix “ting” and “o” in between the word. The query can have mixed workload and can be answer suitably by a combination of B+-Trees on forward and reversed terms or k-gram index.

e. **Boolean Query**

Consider the query “Brutus and Ceaser” it is an example of boolean query. The intent of the user is to get the documents that contains the both the words in a document. The most suitable data structure for such queries are Inverted Index with skip-pointers support, as it can offer quick access to all such terms for intersection.

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

National University of Computer & Emerging Sciences
FAST-Karachi Campus
Information Retrieval (CS317)
Quiz#1

Dated: February 12, 2020 Marks: 20
Time: 20 min.
Std-ID: ______Sol_______

Question No. 1

What are some of the advantages of Boolean Retrieval Model for IR? [5]

Boolean Model of IR is based on Boolean algebra (OR, AND and NOT) operators with features as term present in a document. The presence of exact term in a document qualify for the retrieval based on Boolean logic inferred from the expression.

Drawback

Exact matching may retrieve too few or too many documents
Difficult to rank output, some documents are more important than others
Hard to translate a query into a Boolean expression
All terms are equally weighted

Question No.2

Given the following postings list sizes:

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

What will be the best Boolean Model query processing order for the given query?

Query: (marmalade or skies) and (kaleidoscope or eyes)

marmalade 107913 or skies 271658 = 379571
kaleidoscope 87009 or eyes 213312 = 300321
We will do conservative estimate of the length of the union of postings lists, the recommended order is:
(kaleidoscope or eyes) and (marmalade or skies)

Question No.3

Comments on the following statement as TRUE or FALSE with justification (1-2 line explanations). 5 X 2 marks each.

1. **Stemming increases, the size of the lexicon(dictionary).**

   FALSE. Stemming maps derivational /inflectional morphemes to root word suing heuristic based algorithms hence does not increase size of the dictionary. It is in fact reduce the size of lexicon.

2. **Lemmatization produce human readable features.**

   TRUE. In lemmatization an external lexicon/ thesaurus/dictionary is used and every token from the document is retrain for indexing if it is a dictionary word or lexeme or its variation. Hence the tokens are human readable and understandable, in fact these are lemmas from dictionary.

3. **Extended bi-words index entries, are not true bi-words.**

   TRUE. In an extended bi-words index, the two features (lexical terms) are not necessary the adjacent one, for example: from the text "100 dollars were in my pocket" we can extract "dollars pocket" as a bi-words, which is not a true bi-words of the text.

4. **Positional Index is good for general phrase queries.**

   TRUE. A general phrase query is of the form "w1 w2 …wn". A positional index can easily be used to fetch documents that contains these words from the query in the consecutive positions in a single document. Hence, it can accurately determine the presence of the general phrase.

5. **K-gram index cannot be used to support wildcard queries.**

   TRUE. k-gram index contains is built on extracting all possible k-grams from all the tokens. If a wildcard query of the form "mon*" is executed using a 2-grams index it will have retrieved documents for bi-grams "mo" and "on" etc. There will be a documents that contains "moon" as a feature in the result-set. Obviously there will be false positives.