1111111111111111111111111111111111111111111111111111111111

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

Dated: February 15, 2018 Marks: 30
Time: 25 min.
Std-ID: _____

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 al 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 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

Compare and contrast the following pairs of terms. [10]

Stemming	Lemmatization
It is a heuristic- rule based approach, generally fast and use a single term. It generates unreadable tokens. Stemming algorithms err on the side of being too aggressive, sacrificing precision in order to increase recall.	It is a rigor process that uses a dictionary and uses context to determine the lemma, considered a slow approach. It generates readable lexeme from the dictionary. Lemmatization offers better precision than stemming, but at the expense of recall.
Dictionary	**Thesaurus**
A dictionary contains an alphabetical list of words that includes the meaning, etymology and pronunciation. Organization of words in dictionary in lexicographic order. Dictionary is used to see the meaning, type and pronunciation of word. Dictionary may show use of the word in a sentence.	A thesaurus is a book that contains relationships between words like: synonyms and antonyms. Organization of words in thesaurus in generally in thematic order(conceptual order). Thesaurus is used to see the similarity and differences between pair of words or groups. Thesaurus may show the right usage or different context or sense of words.
Bi-word Index	**Extended Bi-word Index**
Bi-word index contains consecutive words in ordering of their appearance in the document. Example: “Standford university palo alto” may be index as three bi-words like “standford university” “university palo” and “palo alto”.	Extended Bi-word index may contain non-consecutive words generally in order of appearance in the document. Example: Coin in the pocket may be index as “coin pocket”

Question No. 4

1. **Assume a bi-word index is used in an IR system. Give an example of a document which will be returned for a query of "National University FAST" but is actually a false positive which should not be returned. Suggest a solution without a false positive solution. [5]**

   Consider the following two text document:

   D1: The leading computer science institution National University formally called University FAST, was very instrumental in fostering computer culture in Pakistan.

   D2: National University FAST is a leading computer science school in Pakistan.

   It is clearly evident that D2 contains the query text and is the true positive of this query. If a system uses bi-word index it will hit D1 as well which is a false positive the answer to this problem is using a positional indexing to support this type of long phrase query.

2. **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. [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 the both 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.