

National University of Computer & Emerging Sciences FAST-Karachi Campus CS4051- Information Retrieval Quiz#1

Dated: February 21, 2024		Marks: 20
Time: 20 min.		
Std-ID:	Sol	

Question No. 1

What do we mean by Extended Boolean Retrieval Model? Which drawbacks of Boolean IR model it addresses? [5]

The goal of the Extended Boolean Model (EBM) is to overcome the drawbacks of the Boolean model that has been used in information retrieval. The two issues that EBM really solved are as below:

- 1. The Boolean model doesn't consider term weights in queries, and the result set of a Boolean query is often either too small or too big. It tries to incorporate the frequency of the term as a weight.
- 2. The Boolean model gives a flat results- all qualifying documents at the same level (rank or related). It gives higher rank to documents based on the frequency of query terms present in it.

Question No.2

Write down the entries in the permuterm index dictionary that are generated by the terms { "mama", "mammy", "mamy"} [5]

\$mama; \$mammy; \$mamy; a\$mam; ama\$m; ammy\$m; amy\$m; ma\$ma; mama\$; mammy\$

mamy\$; mmy\$ma; my\$mam; y\$mam; y\$mamm

Question No.3

Explain how an IR system can be built to achieve the following evaluation metrics. [5]

a. Recall 100%

We know that Recall = (relevant-retrieved)/ (total-relevant) ------ eq(A), if someone try the best possible effort to get only one relevant document and only return that document as a response to the query it will be Recall = 1/1, mean 100% recall, Obviosuly this kind of system is of no use for the user.

b. Precision 100%

We know that precision = (relevant-retrieved) / (result-set) ------ eq(B), if someone simply return all documents against a given query, the system's Precision = n/n, mean 100% recall as all relevant documents from the collection is returned, Obviously this kind of system is of no use for the user.

Question No.4

Explain the following type of queries from an Information Retrieval prospective with an example. Suggest suitable data structures that get the required answer without false positive. [5]

a. Trailing Wildcard Query

A trailing wildcard query is of the form "mon*". The ideal data structure for retrieving this type of query is to build a B-Tree on the dictionary. It will them search for m, o and n to get all terms that starts on "mon*" You will get 'monday, monkey etc', There will be no false positive in this processing.

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 without false positives in the result-set hence no post processing required.