Name - Vishal Adhav
BU ID - B01007320

Homework 4

I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating, I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment and my grade will be reduced by one level (e.g., from A to A- or from B+ to B) for my first offence, and that I will receive a grade of "F" for the course for any additional offence of any kind.

1.
ANS:

All of the scenarios are possible.
Reason:
Precision = 1:
It is possible if every item which is retrieved is relevant.
But not every relevant item has to be retrieved.

Recall = 1:
It might be possible to get all items even if a few non relevant items are gathered well.

Both = 1:
It is Possible in a best case scenario where the system retrieves all relevant items and no irrelevant items.

2.
ANS:

Document Term Frequency Table

Term	Doc 1	Doc 2	Doc 3	Doc 4
car	1	2	2	1
win	1	1	1	1
race	1	0	1	1

Name - Vishal Adhav
BU ID - B01007320

fast	1	1	0	0
more	1	0	0	0
sport	0	1	0	0
and	0	1	0	0
major	0	0	1	1
formula	0	0	0	1
bet	0	0	0	1
drive	0	0	0	0

Query Term Frequency Table

Term	Query 1	Query 2
car	0	0
win	0	0
race	0	0
fast	0	1
more	0	0
sport	0	1
and	0	0

Name - Vishal Adhav
BU ID - B01007320

major	0	0
formula	1	0
bet	0	0
drive	0	1

Cosine Similarity
Cosine Similarity (d1,d2) = Dot product(d1,d2) / ||d1|| * ||d2||

d1.d2 is the dot product of the vectors
||d1|| & ||d2|| are the magnitudes of vectors

Query / Document	**Document 1**	**Document 2**	**Document 3**	**Document 4**
Formula	**0.0**	**0.0**	**0.0**	**0.4082**
Sport Fast Drive	**0.2582**	**0.4082**	**0.0**	**0.0**

3.
ANS:
Storing the term indexes in a sorted manner which can improve search efficiency by enabling binary search, efficient range queries, better data compression, and faster index construction. This results in quicker search and less storage space used.

Example:
Subject 1: "Database Systems"
Subject 2: "Web Information Retrieval"
Subject 3: "Applied Data Mining"

Name - Vishal Adhav
BU ID - B01007320

Subject 4: "Information Security"

Unsorted Index:

Unsorted Term Index

Term	Document IDs
Database	1
Systems	1
Web	2
Information	2, 4
Retrieval	2
Applied	3
Data	3
Mining	3
Security	4

If we take an example of this unsorted term index, suppose we want to find term Applied, it will take some time to complete the operation and then it will give you the result

Sorted Term Index

Term	Document IDs

Name - Vishal Adhav
BU ID - B01007320

Applied	3
Data	3
Database	1
Information	2, 4
Mining	3
Retrieval	2
Security	4
Systems	1
Web	2

In the sorted term index scenario , we can use binary search example to find the term Applied and above table terms are sorted. We can retrieve the results more quicker than the unsorted term index.

4.
ANS:

No, stemming is not all the time beneficial in all the situation because of its potential to over stem and under stem terms which can then reduce precision and recall in search results:
Over-stemming: When terms that are not mutually exclusive are reduced to the same root, this is known as over-stemming. While more documents are retrieved as a result, which might improve recall, it can also decrease accuracy because the documents may not be pertinent to the query. When "universe" and "university" are stemmed to "univers", for instance, irrelevant results may be returned.

Under-stemming: Sometimes, words that should have the same root are not reduced properly, which can result in missed relevant documents, lowering overall recall.

Example:
Document 4 : “Formula race win major car bet”
Query : “Web information retrieval forms”

Issue: The term "forms" in the query can be reduced to "form" by applying stemming and matching "Formula" in the document.

While the query seeks information on methods or standards in web information retrieval, stemming leads to the inclusion of an irrelevant document about car racing and betting due to superficial word similarity.

5.
ANS:

1) Document 1: fast: 1, car: 1, win: 1, more: 1, race: 1
Vector : [1,1,1,1,1]
Document 4: formula: 1, race: 1, win: 1, major: 1, car: 1, bet: 1
Vector : [1,1,1,1,1,1]
Query 2 TF: sport: 1, fast: 1, drive: 1
Vector: [1, 1, 1]
Dot product - 1
Norm : Query = 1.732
Norm : Doc = 2.236
Cosine similarity : 0.258

2) Documents :
Document 1: "fast car win more race"
Document 2: "sport car and fast car win"
Document 3: "car win major car race"
Document 4: "formula race win major car bet"
Query: “sport fast drive”
formula = TF(t, d) = 1 + log

Document 1 :
sport : 0
Fast : 1 + log(1) = 1
Drive : 0

For Document 2 :

Name - Vishal Adhav
BU ID - B01007320

sport : 1 + log(1) = 1
Fast : 1 + log(2) = 1.693
Drive : 0

Document 3 :
sport : 0
fast : 0
drive : 0

Document 4:
sport : 0
fast : 0
drive : 0

IDF = log(N / number of doc contain t)

Sport : 1.386
fast : 0.693
Drive : 0

TF IDF score = TF * IDF
Doc 1 -
Sport : 0
Fast : 0.693
Drive : 0

Doc 2 -
Sport : 1.386
Fast : 1.174
Drive : 0

Doc 3 -
Sport : 0
Fast : 0
Drive : 0

Doc 4 -
Sport : 0
Fast : 0
Drive : 0

3)
Rankings:

Name - Vishal Adhav
BU ID - B01007320

1. Doc 2 - sport and fast
2. Doc 1 - fast
3. Doc 3 - 0
4. Doc 4 - 0

4)
Due to its simplicity, using natural (raw) frequency for document retrieval might result in errors since it usually gives terms that appear commonly undue weight without taking consideration of their importance across the document data collection.

Example:
Doc 1 : “the the the sky”
Doc : “the sky is beautiful”

Query : “sky”

Problem:
In document 1 , “the” is repeated lot of times and “sky” not repeated
In document 2, “the” appeared once and sky too, Document 2 provides more meaningful content related to the query.
The overuse of words in the document can disturb the result.