CS 429 - Information Retrieval

Homework 4

Shiva Sankar Modala(A20517528)

1 Recitation Exercises

These exercises are to be found in: Introduction to Information Retrieval 1 st Edition (Online) by Christopher Manning, Prabhakar Raghavan, Hinrich Sch¨utze.

1.1 Chapter 7

Exercises: 7.1

We suggested above (Figure 7.2) that the postings for static quality ordering be in decreasing order of g(d). Why do we use the decreasing rather than the increasing order?

Ans: We use the decreasing order rather than the increasing order because this allows us to perform the postings intersection algorithm, i.e., all postings are ordered by a single common ordering. The values of g(d) correspond to the scores (higher g(d) means higher score) and we want decreasing scores when we do retrieval (higher scores should be retrieved first, at the beginning).

1.2 Chapter 8

Exercise 8.1

An IR system returns 8 relevant documents, and 10 nonrelevant documents. There are a total of 20 relevant documents in the collection. What is the precision of the system on this search, and what is its recall?

Ans: We know that,

Precision = # of relevant items retrieved / # retrieved items

Precision = 8 / 18 = 0.444

Recall = #relevant items retrieved / # relevant items

Recall = 8 / 20 = 0.4

Exercise 8.4

What are the possible values for interpolated precision at a recall level of 0?

Ans: We know that,

Pinterp(r) = maxp(r') where r' >= r

Since r = 0, precision could be between 0 and 1 since it must be greater than r 0 < = precision < = 1

Exercise 8.8

Consider an information need for which there are 4 relevant documents in the collection. Contrast two systems run on this collection. Their top 10 results are judged for relevance as follows (the leftmost item is the top ranked search result):

System 1	R N R N N	N N N R R
System 2	N R N N R	R R N N N

a. What is the MAP of each system? Which has a higher MAP?

Ans: MAP(Q) = mean of precision values for individual information needs. Begin with 1/Q which is 1/4 since Q is the number of relevant docs. Then, at each ‘R’ in the list (relevant doc) and do 1/m where m is the number of the doc out of 10 you are on (10 being the total returned number of docs).
MAP of System 1 -> (1/4)*(1/1 + 2/3 + 3/9 + 4/10) = 0.6
MAP of System 2 -> (1/4)*(1/2 + 2/5 + 3/6 + 4/7) = 0.4929
Hence, System 1 has higher MAP.

b. Does this result intuitively make sense? What does it say about what is important in getting a good MAP score?
Ans: This result does make sense because if you look at the distributions of when relevant documents occur in the lists, system 1 has more near the very beginning which contributes to a higher MAP score.

c. What is the R-precision of each system? (Does it rank the systems the same as MAP?)
Ans: R-precision of System 1 -> 2/4 = 1/2
R-precision of System 2 -> 1/4
The ranking is the same as MAP, with system 1 having a higher number.

Exercise 8.10
Below is a table showing how two human judges rated the relevance of a set of 12 documents to a particular information need (0 = nonrelevant, 1 = relevant). Let us assume that you’ve written an IR system that for this query returns the set of documents {4, 5, 6, 7, 8}.

docID	Judge 1	Judge 2
1	0	0
2	0	0
3	1	1
4	1	1
5	1	0
6	1	0
7	1	0
8	1	0
9	0	1
10	0	1
11	0	1

12	0	1

a. Calculate the kappa measure between the two judges.
Ans: Kappa = (P(A) - P(E)) / (1-P(E))
P(A) = the proportion of the times the judges agreed = 4/12
P(E) = the number of times they would be expected to agree by chance, since 2 judges this is 0.5 (6/12)
Kappa = (4/12 - 6/12) / (1-6/12) = -0.3333

b. Calculate precision, recall, and F1 of your system if a document is considered relevant only if the two judges agree.
Ans: We’re only looking at documents {4, 5, 6, 7, 8}.
Precision = # of relevant items retrieved / # retrieved items
Precision = 1/5 (only one of the 5 retrieved is relevant)
Recall = #relevant items retrieved / # relevant items
Recall = 1/2 (only 1 of the 4 relevant (docs 3 and 4) are retrieved)
F1 = 2PR / (P+R) F1 = (2*1/5*1/2) / (1/5 +1/2) = 2/7 = 0.2857

c. Calculate precision, recall, and F1 of your system if a document is considered relevant if either judge thinks it is relevant.
Ans: Precision = # of relevant items retrieved / # retrieved items
Precision = 5/5 = 1 (all 5 of the retrieved are relevant)
Recall = #relevant items retrieved / # relevant items
Recall = 5/10 = 1/2 (5 of the 10 relevant docs are retrieved)
F1 = 2PR / (P+R) F1 = (2*1*1/2) / (1 +1/2) = 2/3 = 0.6666

1.3 Chapter 9
Exercise 9.1
In Rocchio’s algorithm, what weight setting for α/β/γ does a “Find pages like this one” search correspond to?
Ans: γ = 0 because only positive feedback is allowed.
α = 0 because it’s not using the query vector at all
β = 1 because that’s the only option with the other constraints

Exercise 9.3
Under what conditions would the modified query qm in Equation 9.3 be the same as the original query q0? In all other cases, is qm closer than q0 to the centroid of the relevant documents?

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j \qquad (9.3)$$

Ans: If the second and third parts of the equation are equal in absolute value (the part with beta and the relevant docs, and the part with y and the non-relevant docs) then you will just be left with alpha times the query vector. If alpha is 1, qm = q0.
No, it’s not true that qm is closer than q0 to the centroid of the relevant docs. It would depend on the parameter values. Though the point is to move towards the centroid of relevant docs, may not always be the case. We want to use reasonable parameter values to achieve this, but it's possible to not use reasonable values and make it not move to that centroid of relevant docs (in which case the original could be closer)

Exercise 9.5
Suppose that a user’s initial query is cheap CDs cheap DVDs extremely cheap CDs. The user examines two documents, d1 and d2. She judges d1 , with the content CDs cheap software cheap CDs relevant and d2 with content cheap thrills DVDs nonrelevant. Assume that we are using direct term frequency (with no scaling and no document frequency). There is no need to length-normalize vectors. Using Rocchio relevance feedback as in Equation (9.3) what would the revised query vector be after relevance feedback? Assume α = 1, β = 0.75, γ = 0.25.

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \tag{9.3}$$

Ans: Q: cheap CDs cheap DVDs extremely cheap CDs
D1: CDs cheap software cheap CDs
D2: cheap thrills DVDs

Word -> number in d1, number in d2, number in query
CDs -> 2 0 2
Cheap -> 2 1 3
Software -> 1 0 0
Thrills -> 0 1 0
Extremely -> 0 0 1
DVDs -> 0 1 1

Qm = 1*query + 0.75*d1 - 0.25*d2
= [2, 3, 0, 0, 1, 1] + [1.5, 1.5 , 0.75, 0, 0, 0] - [0, 0.25, 0, 0.25, 0, 0.25]
= [3.5, 4.25, 0,75, -0.25 (set this to zero), 1, 0.75]

Exercise 9.6
Omar has implemented a relevance feedback web search system, where he is going to do relevance feedback based only on words in the title text returned for a page (for efficiency). The user is going to rank 3 results. The first user, Jinxing, queries for:

banana slug

and the top three titles returned are:

banana slug Ariolimax columbianus

Santa Cruz mountains banana slug
Santa Cruz Campus Mascot

Jinxing judges the first two documents relevant, and the third nonrelevant. Assume that Omar’s search engine uses term frequency but no length normalization nor IDF. Assume that he is using the Rocchio relevance feedback mechanism, with α = β = γ = 1. Show the final revised query that would be run. (Please list the vector elements in alphabetical order.)

Ans: Vector order -> query, d1, d2, d3
Ariolimax: 0 1 0 0
Banana: 1 1 1 0
Campus: 0 0 0 1
Columbianus: 0 1 0 0
Cruz: 0 0 1 1
Mascot: 0 0 0 1
Mountains: 0 0 1 0
Santa: 0 0 1 1
Slug: 1 1 1 0

0 + ½*1 + ½*0 - 0 = ½
1 + ½*1 + ½*1 - 0 = 2
0 + ½*0 + ½*0 - 1 = -1
0 + ½*1 + ½*0 - 0 = ½
0 + ½*0 + ½*1 - 1 = -½
0 + ½*0 + ½*0 - 1 = -1
0 + ½*0 + ½*1 - 0 = ½
0 + ½*0 + ½*1 - 1 = -½
1 + ½*1 + ½*1 - 0 = 2

So -> [½, 2, 0, ½, 0, 0, ½, 0, 2]

Exercise 9.7
If A is simply a Boolean cooccurrence matrix, then what do you get as the entries in C?
Ans: C = AA^Transpose, so every entry in C is the number of times that two terms show up together (number of docs where they co-occur).

1.4 Chapter 10
Exercises: N/A