Homework 2

Page 110: Exercise 6.10; Exercise 6.12
Page 116: Exercise 6.15; Exercise 6.17
Page 121: Exercise 6.19
Page 122: Exercise 6.20; Exercise 6.23; Exercise 6.24
Page 131: Exercise 7.3; Exercise 7.5; Exercise 7.8
Page 144: Exercise 8.1
Page 145: Exercise 8.3
Page 150: Exercise 8.9
Page 154: Exercise 8.10
Page 167: Exercise 9.3
Page 177: Exercise 9.7
Page 211: Exercise 11.3
Page 228: Exercise 12.6; Exercise 12.7

Exercise 6.10

Consider the table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3 in Figure 6.9. Compute the tf-idf weights for the terms car, auto, insurance, best, for eachdocument, using the idf values from Figure 6.8.

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

► **Figure 6.9** Table of tf values for Exercise 6.10.

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

► **Figure 6.8** Example of idf values. Here we give the idf's of terms with various frequencies in the Reuters collection of 806,791 documents.

Solution

	Doc1	Doc2	Doc3
car	44.55	6.6	39.6
Auto	6.24	68.64	0
Insurance	0	53.46	46.98

Best	21	0	25.5

Exercise 6.12
How does the base of the logarithm in (6.7) affect the score calculation in (6.9)? How does the base of the logarithm affect the relative scores of two documents on a given query?

Solution

For any base b > 0, $\text{idf(b)} = \log_b(\frac{N}{df}) = \frac{\log_{10}(\frac{N}{df})}{\log_{10}(b)}$,

$\frac{\text{idf(b)}}{\text{idf(10)}} = \frac{1}{\log_{10}(b)}$is a constant.

So changing the base affects the score by a factor $\frac{1}{\log_{10}(b)}$, and the relative scores of two documents on a given query are not affected.

Exercise 6.15
Recall the tf-idf weights computed in Exercise 6.10. Compute the Euclidean normalized document vectors for each of the documents, where each vector has four components, one for each of the four terms.

Solution
doc1 = [0.8974, 0.1257, 0, 0.4230]
doc2 = [0.0756, 0.7867, 0.6127, 0]
doc3 = [0.5953, 0, 0.7062, 0.3833]

Exercise 6.17
With term weights as computed in Exercise 6.15, rank the three documents by computed score for the query car insurance, for each of the following cases of term weighting in the query:
1. The weight of a term is 1 if present in the query, 0 otherwise.
2. Euclidean normalized idf.

Solution
1. q = [1, 0, 1, 0]
score(q, doc1)= 0.8974, score(q, doc2) = 0.6883, score(q, doc3) = 1.3015
Ranking: doc3, doc1, doc2
2. q = [0.4778, 0.6024, 0.4692, 0.4344]
score(q, doc1) = 0.6883, score(q, doc2) = 0.7975, score(q, doc3) = 0.7823
Ranking: doc2, doc3, doc1

Exercise 6.19
Compute the vector space similarity between the query “digital cameras” and the

document “digital cameras and video cameras” by filling out the empty columns in Table 6.1. Assume N = 10,000,000, logarithmic term weighting (wf columns) for query and document, idf weighting for the query only and cosine normalization for the document only. Treat and as a stop word. Enter term counts in the tf columns. What is the final similarity score?

Solution

Word	Query					document			qi*di
	tf	wf	df	idf	qi=wf-idf	tf	wf	di=normalized wf	
digital	1	1	10,000	3	3	1	1	0.52	1.56
video	0	0	100,000	2	0	1	1	0.52	0
Cameras	1	1	50,000	2.3	2.3	2	1.3	0.68	1.56

Similarity score = 1.56+1.56 = 3.12

Exercise 6.20

Show that for the query **affection**, the relative ordering of the scores of the three documents in Figure 6.13 is the reverse of the ordering of the scores for the query **jealous gossip**.

Solution:

For the query **affection**, score(q, SaS) = 0.996, score(q, PaP) = 0.993, score(q, WH) = 0.847, so the order is SaS, PaP, WH.

For the query **jealous gossip**, score(q, SaS) = 0.104, score(q, PaP) = 0.12, score(q, WH) = 0.72, so the order is WH, PaP,SaS.

So the latter case is the reverse order of the former case.

Exercise 6.23

Refer to the tf and idf values for four terms and three documents in Exercise 6.10. Compute the two top scoring documents on the query **best car insurance**for each of the following weighing schemes: (i) `nnn.atc`; (ii) `ntc.atc`.

Solution

(i) nnn.atc

nnn weights for documents

Term	Doc1	Doc2	Doc3
Car	27	4	24
Auto	3	33	0
Insurance	0	33	29
Best	4	0	17

	Query				Product		
Term	tf(augmented)	idf	tf-idf	atc weight	Doc1	Doc2	Doc3

Car	1	1.65	1.65	0.56	15.12	2.24	13.44
Auto	0.5	2.08	1.04	0.353	1.06	11.65	0
Insurance	1	1.62	1.62	0.55	0	18.15	15.95
Best	1	1.5	1.5	0.51	7.14	0	8.67

Score(q, doc1) = 15.12 + 1.06 +0 + 7.14 = 23.32, score(q, doc2) = 2.24 + 11.65 + 18.15 + 0 = 32.04, score(q, doc3) = 13.44 + 0 + 15.95 + 8.67 = 38.06
Ranking: doc3, doc2, doc1

(ii) ntc.atc
ntc weight for doc1

Term	tf(augmented)	Idf	tf-idf	Normalized weights
Car	27	1.65	44.55	0.897
Auto	3	2.08	6.24	0.125
Insurance	0	1.62	0	0
Best	14	1.5	21	0.423

ntc weight for doc2

Term	tf(augmented)	Idf	tf-idf	Normalized weights
Car	4	1.65	6.6	0.075
Auto	33	2.08	68.64	0.786
Insurance	33	1.62	53.46	0.613
Best	0	1.5	0	0

ntc weight for doc3

Term	tf(augmented)	idf	tf-idf	Normalized weights
Car	24	1.65	39.6	0.595
Auto	0	2.08	0	0
Insurance	29	1.62	46.98	0.706
Best	117	1.5	25.5	0.383

	query				Product		
Term	tf(augmented)	idf	tf-idf	atc weight	Doc1	Doc2	Doc3
Car	1	1.65	1.65	0.56	0.502	0.042	0.33
Auto	0.5	2.08	1.04	0.353	0.044	0.277	0
Insurance	1	1.62	1.62	0.55	0	0.337	0.38
Best	1	1.5	1.5	0.51	0.216	0	0.19

Score(q, doc1) = 0.762, score(q, doc2) = 0.657, score(q, doc3) = 0.916
Ranking: doc3, doc1, doc2

Exercise 6.24
Suppose that the word **coyote** does not occur in the collection used in Exercises 6.10

and6.23. How would one compute `ntc.atc`scores for the query **coyote insurance**?

Solution
For the ntc weight, we compute the ntc weight of insurance.
For the atc weight, there is no need to compute, because the ntc weight for all documents is 0 for coyote.

Exercise 7.3
If we were to only have one-term queries, explain why the use of global champion lists with $r = K$ suffices for identifying the K highest scoring documents. What is a simple modification to this idea if we were to only have s-term queries for any fixed integers >1?

Solution
1. We take the union of the champion lists for each of the terms comprising the query, and restrict cosine computation to only the documents in the union set. If the query contains only one term, we just take the list with r = K, because there is no need to compute the union.
2. For each term, identify the $\left\lceil \frac{K}{s} \right\rceil$ highest scoring documents.

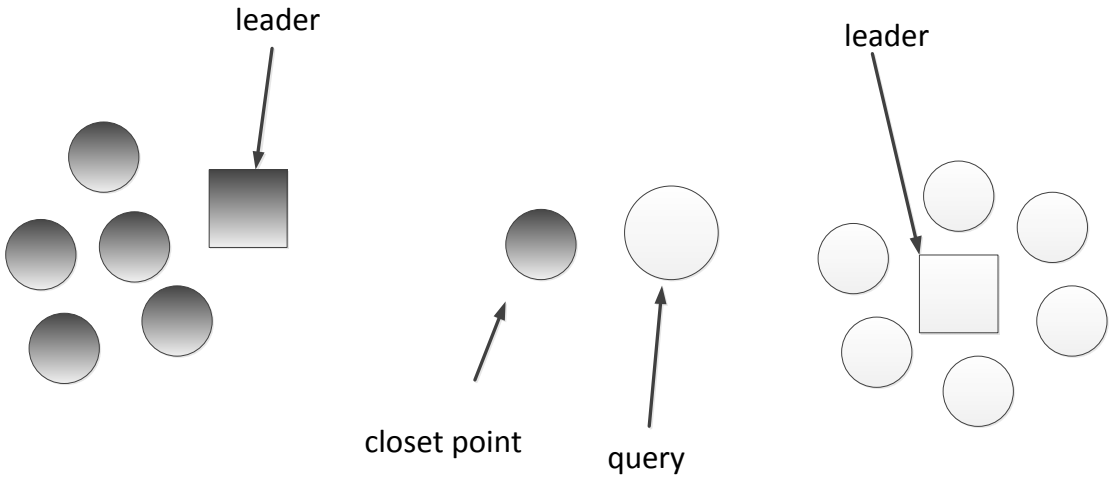
Exercise 7.5
Consider again the data of Exercise 6.23 with `nnn.atc`for the query-dependent scoring. Suppose that we were given static quality scores of 1 for Doc1 and 2 for Doc2.Determine under Equation (7.2) what ranges of static quality score for Doc3 result in it being the first, second or third result for the query best car insurance.

Solution
Suppose the static quality score for Doc3 is g(doc3).
According to Exercise 6.23 and Equation 7.2, score(doc1, q) = 0.7627+ 1 = 1.7627, score(doc2, q) = 0.6839 + 2 = 2.6839, score(doc3, q) = 0.9211 + g(doc3).
For Doc3 result in being:
(1) the first: 0.9211 + g(doc3) > 2.6839, we get g(doc3) > 1.7628
(2) the second: 1.7627< 0.9211 + g(doc3) < 2.6839, we get 0.8416 < g(doc3) < 1.7628
(3) the third: 0.9211+ g(doc3) <1.7627, we get 0 <=g(doc3) < 0.8416.

Exercise 7.8
The nearest-neighbor problem in the plane is the following: given a set of N data points on the plane, we preprocess them into some data structure such that, given a query point Q, we seek the point in N that is closest to Q in Euclidean distance. Clearly cluster pruning can be used as an approach to the nearest-neighbor problem in the plane, if we wished to avoid computing the distance from Q to every one of the query points. Devise a simple example on the plane so that with two leaders, the

answer returned by cluster pruning is incorrect (it is not the data point closest to *Q*).

Solution



As is shown in the above picture, the right leader is closer to the query point than the left leader, but the closest point belongs to the left group.

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?

Solution
Precision = 8/18 = 0.44
Recall = 8/20 = 0.4

Exercise 8.3
Derive the equivalence between the two formulas for F measure shown in Equation (8.5), given that $\alpha = 1/(\beta^2 + 1)$.

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{1}{\frac{1}{\beta^2+1}\frac{1}{P} + \frac{\beta^2}{\beta^2+1}\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

Exercise 8.9
The following list of Rs and Ns represents relevant (R) and nonrelevant (N) returned documents in a ranked list of 20 documents retrieved in response to a query from a collection of 10,000 documents. The top of the ranked list (the document the system thinks is most likely to be relevant) is on the left of the list. This list shows 6 relevant documents. Assume that there are 8 relevant documents in total in the collection.
R R N NNNNN R N R N NN R N NNN R

a. What is the precision of the system on the top 20?
b. What is the F1 on the top 20?
c. What is the uninterpolated precision of the system at 25% recall?
d. What is the interpolated precision at 33% recall?
e. Assume that these 20 documents are the complete result set of the system. What is the MAP for the query?
Assume, now, instead, that the system returned the entire 10,000 documents in a ranked list, and these are the first 20 results returned.
f. What is the largest possible MAP that this system could have?
g. What is the smallest possible MAP that this system could have?
h. In a set of experiments, only the top 20 results are evaluated by hand. The result in (e) is used to approximate the range (f)–(g). For this example, how large (in absolute terms) can the error for the MAP be by calculating (e) instead of (f) and (g) for this query?

Solution

a. Precision = 6/20 = 0.3
b. Recall = 6/8 = 0.75

$$F_1 = \frac{2PR}{P+R} = \frac{3}{7} = 0.43$$

c. 8*0.25 = 2, the uninterpolated precision could be 1, 2/3, 2/4, 2/5, 2/6, 2/7, 1/4
d. Because the highest precision found for any recall level larger than 33% is 4/11 = 0.364, hence the interpolated precision at 33% recall is 4/11 = 0.364.

e. MAP = 1/6 * (1 + 1 + 3/9 + 4/11 + 5/15 + 6/20) = 0.555

f. $\text{MAP}_{\text{largest}} = \frac{1}{8} * \left(1 + 1 + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20} + \frac{7}{21} + \frac{8}{22}\right) = 0.503$

g. $\text{MAP}_{\text{smallest}} = \frac{1}{8} * \left(1 + 1 + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20} + \frac{7}{9999} + \frac{8}{10000}\right) = 0.417$

h. 0.555-0.417 = 0.138, 0.555-0.503 = 0.052
the error is in [0.052, 0.138]

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.
b. Calculate precision, recall, and F1 of your system if a document is considered relevant only if the two judges agree.
c. Calculate precision, recall, and F1 of your system if a document is considered relevant if either judge thinks it is relevant.

Solution
(a)
P(A) = 4/12 = 1/3
P(nonrelevant) = (6+6)/(12+12) = 0.5, P(relevant) = (6+6)/(12+12) = 0.5
P(E)=0.5*0.5 + 0.5*0.5 = 0.5
Kappa = (P(A)-P(E))/(1-P(E)) = -1/3
(b)
Precision = 1/5 = 0.2
Recall = 1/2 = 0.5
F1 = 2*0.2*0.5/(0.2+0.5) = 2/7 = 0.286
(c)
Precision = 5/5 = 1
Recall = 5/10 = 0.5
F1 = 2*1*0.5/(1+0.5) = 2/3 = 0.667

The Exercise 9.3 on the book and the pdf edition are different.

Exercise 9.3 (pdf edition)

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?

Solution

$$q_m = \alpha q_0 + \beta \frac{1}{|D_r|} \sum_{d_j \in D_r} d_j - \gamma \frac{1}{|D_{nr}|} \sum_{d_j \in D_{nr}} d_j = q_0$$

One possible condition is as follows:

$\alpha = 1,\ \sum_{d_j \in D_r} d_j = k \sum_{d_j \in D_{nr}} d_j\,, k\ is\ constant.$

No, if β is very small and γ is very large, q_0 might be closer to the centroid of the relevant documents than q_m .

Exercise 9.3 (book)

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 *alpha*= 1, *beta*= 0.75, *gamma*= 0.25.

Solution

	cheap	CDs	DVDs	extremely	software	thrills
q_0	3	2	1	1	0	0
d_1	2	2	0	0	1	0
d_2	1	0	1	0	0	1

Using Rocchio algorithm, $q_m=q_0 + 0.75*d_1-0.25*d2$. Negative weights are set to 0.

q_m	4.25	3.5	0.75	1	0.75	0

Exercise 9.7

If *A* is simply a Boolean "incidence" matrix, then what do you get as the entries in *C*?

Solution

$C_{u,v}$ is the number of documents containing both term u and term v.

By Boolean co-occurrence matrix, it means the element $A_{t,d}$, term t appears in document d. As C is the similarity score, we increase $C_{u,v}$ by one if term u and term v both appear in the document .

If $A_{u,d}\ AND\ A_{v,d} = 1, C_{u,v} = C_{u,v} + 1$.

Exercise 11.3

Let Xt be a random variable indicating whether the term *t* appears in a document. Suppose we have $|R|$ relevant documents in the document collection and that $Xt = 1$ ins of the documents. Take the observed data to be just these observations of Xt for each document in *R*. Show that the MLE for the parameter $pt = P(Xt= 1 \mid R = 1,q)$, that is, the value for pt which maximizes the probability of the observed data, is $pt = s/|R|$.

Solution

$P(D|t) = \prod_{d\in D} P(d|t) = \theta^n(1-\theta)^{N-n}$, where $\theta = P(X_t = 1|R = 1, q)$, n is the number of documents containing the term, and N is the total number of documents.

The log-likelihood: $\log P(D|t) = n \log\theta + (N-n) \log\theta$

Let the partial derivative equal to 0, $\frac{\partial \log P}{\partial \theta} = \frac{n}{\theta} - \frac{N-n}{1-\theta} = 0$.

We get $\theta = \frac{n}{N}$, so $p_t = s/|R|$.

Exercise 12.6

Consider making a language model from the following training text:

The martian has landed on the latin pop sensation ricky martin

a. Under a MLE-estimated unigram probability model, what are *P*(the) and *P*(martian)?

b. Under a MLE-estimated bigram model, what are *P*(sensation | pop) and *P*(pop | the)?

Solution

a. P(the) = 2/11, P(martian) = 1/11

b. *P*(sensation | pop) = 1, *P*(pop | the) = 0

Exercise 12.7

Suppose we have a collection that consists of the 4 documents given in the below table.

docID Document text

1 click go the shears boys click click click

2 click click

3 metal here

4 metal shears click here

Build a query likelihood language model for this document collection. Assume a mixture model between the documents and the collection, with both weighted at 0.5. Maximum likelihood estimation (mle) is used to estimate both as unigram models. Work out the model probabilities of the queries click, shears, and hence click shears for

each document, and use those probabilities to rank the documents returned by each query. Fill in these probabilities in the below table:

Query Doc 1 Doc 2 Doc 3 Doc 4

click

shears

click shears

What is the final ranking of the documents for the query click shears?

Solution

Language models

	click	go	the	shears	boys	metal	here
model1	1/2	1/8	1/8	1/8	1/8	0	0
model2	1	0	0	0	0	0	0
model3	0	0	0	0	0	1/2	1/2
model4	1/4	0	0	1/4	0	1/4	1/4

Collection model	7/16	1/16	1/16	2/16	1/16	2/16	2/16

model probabilities of the queries

query	doc1	doc2	doc3	doc4
click	0.4688	0.7188	0.2188	0.3438
shears	0.125	0.0625	0.0625	0.1875
click shears	0.0586	0.0449	0.0137	0.0645

Final ranking for the query click shears: doc4 > doc1 > doc2 > doc3