

Exercise 6.8, 6.9, 6.10

- 6.8: Why is the idf of a term always finite?
- 6.9: What is the idf of a term that occurs in every document? Compare this with the use of stop word lists.
- 6.10: Consider the following table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3. Compute the tf-idf weights for the terms *car*, *auto*, *insurance*, *best*, for each document, using the idf values:
 $\text{idf}(\text{car})=1.65$; $\text{idf}(\text{auto})=2.08$, $\text{idf}(\text{insurance})=1.62$;
 $\text{idf}(\text{best})=1.5$

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

Exercises 6.12, 6.13, 6.14

- 6.12: How does the base of the logarithm affect idf calculation?
- How does the base of the logarithm affect the relative scores of two documents on a given query?
- 6.13: If the logarithm in the idf formula is computed base 2, suggest a simple approximation to the idf of a term.
- 6.14: If we stem jealous and jealousy to a common stem before setting up the vector space, detail how the definitions of tf and idf (for the common stem) should be modified.

Exercises 6.15, 6.17

- 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.
- 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: *car, insurance*
 - 1. The weight of a term is 1 if present in the query, 0 otherwise.
 - 2. Euclidean normalized idf.

Exercise 6.18

- One measure of the similarity of two vectors is the *Euclidean distance* (or L_2 distance) between them:

$$|\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^M (x_i - y_i)^2}$$

- Given a query q and documents d_1, d_2, \dots , we may rank the documents d_i in order of **increasing** Euclidean distance from q .
- Show that if q and the d_i are all normalized to unit vectors, then the rank ordering produced by Euclidean distance is identical to that produced by cosine similarities.

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 below. 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?

| word | query | | | | | document | | | $q_i \cdot d_i$ |
|---------|-------|----|---------|-----|-----------------------|----------|----|------------------------------|-----------------|
| | tf | wf | df | idf | $q_i = \text{wf-idf}$ | tf | wf | $d_i = \text{normalized wf}$ | |
| digital | | | 10,000 | | | | | | |
| video | | | 100,000 | | | | | | |
| cameras | | | 50,000 | | | | | | |

Exercises 7.1, 7.3

- 7.1: We suggested 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?
- 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.

Exercise 7.6

- Sketch the frequency-ordered postings for the data in the following table

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

Exercise 7.7

- Let the static quality scores for Doc1, Doc2 and Doc3 be respectively 0.25, 0.5 and 1. Sketch the postings for impact ordering when each postings list is ordered by the sum of the static quality score and the Euclidean normalized tf values in the following table

| | Doc1 | Doc2 | Doc3 |
|-----------|------|------|------|
| car | 0.88 | 0.09 | 0.58 |
| auto | 0.10 | 0.71 | 0 |
| insurance | 0 | 0.71 | 0.70 |
| best | 0.46 | 0 | 0.41 |