# Multimedia Information Retrieval and Technology

## Lecture 5. Weighting

By : Fangyu Wu

Room: SD555

Xi'an Jiaotong-Liverpool University
西交利物浦大学

What are the two key statistics to describe the effectiveness of an information retrieval system? Give the definitions of these two statistics.

Ranked retrieval

a. Term frequency, document freq, collection freq
b. idf weighting
c. tf-idf weighting

Xi'an Jiaotong-Liverpool University
西交利物浦大學

## Ranked retrieval

Thus far, our queries have all been Boolean.

- Documents either match or don't.

Good for expert users with precise understanding of their needs and the collection.

Not good for the majority of users.

- Most users incapable of writing Boolean queries (or they are, but they think it's too much work).
- Most users don't want to wade through 1000s of results.
  - This is particularly true of web search.

Xi'an Jiaotong-Liverpool University 西交利物浦大學

## Ranked retrieval models

Rather than a set of documents satisfying a query expression, in ranked retrieval, the system returns an ordering over the (top) documents in the collection for a query.

Free text queries: Rather than a query language of operators and expressions, the user's query is just one or more words in a human language

Xi'an Jiaotong-Liverpool University
西交利物浦大學

## Feast or famine: not a problem in ranked retrieval

When a system produces a ranked result set, large result sets are not an issue

- We just show the top *k* ( ≈ 10) results
- We don't overwhelm the user

- Premise: the ranking algorithm works

## Scoring as the basis of ranked retrieval

We wish to return in order the documents most likely to be useful to the searcher

- How can we rank-order the documents in the collection with respect to a query?

Assign a score – say in [0, 1] – to each document.

This score measures how well document and query "match".

## Query-document matching scores

We need a way of assigning a score to a query/document pair.

Let's start with a one-term query: If the query term does not occur in the document: score should be 0

The more frequent the query term in the document, the higher the score (should be)

## Take 1: Jaccard coefficient

A commonly used measure of overlap of two sets *A* and *B*

jaccard$(A,B) = |A \cap B| / |A \cup B|$

jaccard$(A,A) = 1$

jaccard$(A,B) = 0$ if $A \cap B = 0$

*A* and *B* don't have to be the same size.

Always assigns a number between 0 and 1.

## Jaccard coefficient: Scoring example

What is the query-document match score that the Jaccard coefficient computes for each of the two documents below (if there are no stop words)?

Query: *ides of march*

Document 1: *caesar died in march*

Document 2: *the long march*

## Jaccard coefficient: Scoring example

What is the query-document match score that the Jaccard coefficient computes for each of the two documents below (if there are no stop words)?

Query: *ides of march*

Document 1: *caesar died in march*

Document 2: *the long march*

jaccard$(A,B)$ = $|A \cap B| \,/\, |A \cup B|$

D1: $|A \cap B|$=1, $|A \cup B|$=6, jaccard$(A,B) = 1/6$

*D2:* $|A \cap B|$=1, $|A \cup B|$=5, jaccard$(A,B) = 1/5$

## Exercise

For this problem, consider a simple collection with the following two documents:

**Document 1:** the way to the school is long and hard when walking in the rain
**Document 2:** the rain has not stopped in days and the school has closed

Consider the query:

**Query:** school closed rain

Also, consider the following stop word list: [the, to, is, and, in, has, not]

What are the similarity scores of the query with each document given above using Jaccard coeffcient if there are no stop words? What are the similarity scores if we use the stop words?

Xi'an Jiaotong-Liverpool University
西交利物浦大学

We consider the number of distinct words for the two cases, and use that the Jaccard Coefficient is given by $\frac{|A \cap B|}{|A \cup B|}$ to get:

a.) Without the stopword list (we consider all words in our calculations):

| Doc | $\lvert\text{Doc} \cap \text{Query}\rvert$ | $\lvert\text{Doc} \cup \text{Query}\rvert$ | Jaccard |
|---|---|---|---|
| Doc1 | 2 | 13 | $2/13 = 0.1538$ |
| Doc2 | 3 | 10 | $3/10 = 0.3$ |

b.) With the stopword list (we ignore stopwords in our calculations):

| Doc | $\lvert\text{Doc} \cap \text{Query}\rvert$ | $\lvert\text{Doc} \cup \text{Query}\rvert$ | Jaccard |
|---|---|---|---|
| Doc1 | 2 | 8 | $2/8 = 0.25$ |
| Doc2 | 3 | 5 | $3/5 = 0.6$ |

## Issues with Jaccard for scoring

We need a more sophisticated way of normalizing for length

we'll use

. . . instead of |A ∩ B|/|A ∪ B| (Jaccard) for length normalization.

$$|A \cap B| / \sqrt{|A \cup B|}$$

Xi'an Jiaotong-Liverpool University 西交利物浦大学

## Issues with Jaccard for scoring

- It doesn’t consider *term frequency* (how many times a term occurs in a document)
- Rare terms in a collection are more informative than frequent terms *(document frequency).*

Jaccard doesn’t consider these informations.

I. Ranked retrieval
   a. Term frequency, document freq, collection freq
   b. idf weighting
   c. tf-idf weighting

Xi'an Jiaotong-Liverpool University
西交利物浦大学

## *Bag of words* model

Vector representation doesn't consider the ordering of words in a document.

- *John is quicker than Mary*
- *Mary is quicker than John*

have the same vectors

This is called the bag of words model.

Xi'an Jiaotong-Liverpool University 西交利物浦大学

## Recall (Lecture 1): Binary term-document incidence matrix

| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 1 | 1 | 0 | 0 | 0 | 1 |
| Brutus | 1 | 1 | 0 | 1 | 0 | 0 |
| Caesar | 1 | 1 | 0 | 1 | 1 | 1 |
| Calpurnia | 0 | 1 | 0 | 0 | 0 | 0 |
| Cleopatra | 1 | 0 | 0 | 0 | 0 | 0 |
| mercy | 1 | 0 | 1 | 1 | 1 | 1 |
| worser | 1 | 0 | 1 | 1 | 1 | 0 |

Each document is represented by a binary vector $\in \{0,1\}^{|V|}$

Xi'an Jiaotong-Liverpool University 西交利物浦大学

## Term-document count matrices

Consider the number of occurrences of a term in a document:

Each document is a count vector : a column below

| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 157 | 73 | 0 | 0 | 0 | 0 |
| Brutus | 4 | 157 | 0 | 1 | 0 | 0 |
| Caesar | 232 | 227 | 0 | 2 | 1 | 1 |
| Calpurnia | 0 | 10 | 0 | 0 | 0 | 0 |
| Cleopatra | 57 | 0 | 0 | 0 | 0 | 0 |
| mercy | 2 | 0 | 3 | 5 | 5 | 1 |
| worser | 2 | 0 | 1 | 1 | 1 | 0 |

Xi'an Jiaotong-Liverpool University
西交利物浦大學

## Term frequency

A document or zone that mentions a query term **more often （higher tf）** has more to do with that query and therefore should receive a higher score.

## Term frequency tf

The term frequency $\mathrm{tf}_{t,d}$ of term *t* in document *d* is defined as the number of times that *t* occurs in *d*.

## Exercise

Consider following documents with the stop word list: [when, in, the, and, I]

Doc 1: when walking in the rain

Doc 2: rain stopped walk, I ran, rain stop.

Doc 3: stop walking and run

Determine the term frequency for the term **stop**.

Term frequency: $d_1 = 0, d_2 = 2, d_3 = 1$

Xi'an Jiaotong-Liverpool University
西交利物浦大學

## Term frequency tf

We want to use tf when computing query-document match scores. But how?

Raw term frequency is not what we want:

- A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term.
- But not 10 times more relevant.

Relevance does not increase proportionally with term frequency.

Xi'an Jiaotong-Liverpool University 西交利物浦大学

## Log-frequency weighting

The log frequency weight of term t in d is

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

0 → 0, 1 → 1, 2 → 1.3, 10 → 2, 1000 → 4, etc.

Score for a document-query pair: sum over terms *t* in both *q* and *d*:

$$\text{score} = \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$$

The score is 0 if none of the query terms is present in the document.

## Term frequency tf

Are all words in a collection equally important?

Term frequency suffers a critical problem:

- All terms are considered equally important when it comes to assessing .

## Collection frequency

Rare terms are more informative than frequent terms

- Recall stop words

A document containing this term is very likely to be relevant

- → We want a high weight for rare terms.

## Collection frequency

The **collection frequency** of *t* is the number of occurrences of *t* in the collection, counting multiple occurrences.

An immediate idea is to scale down the term weights of terms with high *collection frequency*.

## Collection vs. Document frequency

Example:

Which word is a better search term (and should get a higher weight)?

| Word | Collection frequency | Document frequency |
|---|---:|---:|
| *insurance* | 10440 | 3997 |
| *try* | 10422 | 8760 |

A document-level statistic

## Collection vs. Document frequency

*Document frequency* defined to be the number of documents in the collection that contain a term *t*.

Example:

Which word is a better search term (and should get a higher weight)?

| Word | Collection frequency | Document frequency |
|---|---|---|
| *insurance* | 10440 | 3997 |
| *try* | 10422 | 8760 |

Xi'an Jiaotong-Liverpool University
西交利物浦大学

## Collection vs. Document frequency

The collection frequency (cf) and document frequency (df) can behave rather differently.

- the cf values for both *try* and *insurance* are roughly equal, but their df values differ significantly.
- Intuitively, we want the few documents that contain insurance to get a higher boost for a query on insurance than the many documents containing try get from a query on try.

| Word | Collection frequency | Document frequency |
|---|---:|---:|
| *insurance* | 10440 | 3997 |
| *try* | 10422 | 8760 |

I. Parametric and zone indexes

II. Ranked retrieval

   a. Term frequency, document freq, collection freq

   b. idf weighting

   c. tf-idf weighting

## idf weight

- $\mathrm{df}_t$ is an inverse measure of the informativeness of *t*
- $\mathrm{df}_t \leq N$

We define the idf (**inverse document frequency)** of *t* by

$$\mathrm{idf}_t = \log_{10}(N/\mathrm{df}_t)$$

- We use log ($N/\mathrm{df}_t$) instead of $N/\mathrm{df}_t$ to “dampen” the effect of idf.

Xi’an Jiaotong-Liverpool University 西交利物浦大学

## idf example, suppose *N* = 1 million

| term | $\text{df}_t$ | $\text{idf}_t$ |
|---|---:|---:|
| calpurnia | 1 | 6 |
| animal | 100 | 4 |
| sunday | 1,000 | 3 |
| fly | 10,000 | 2 |
| under | 100,000 | 1 |
| the | 1,000,000 | 0 |

$$\text{idf}_t = \log_{10}(N/\text{df}_t)$$

Xi'an Jiaotong-Liverpool University
西交利物浦大学

## idf example, suppose *N* = 1 million

| term | $\text{df}_t$ | $\text{idf}_t$ |
|---|---:|---:|
| calpurnia | 1 | 6 |
| animal | 100 | 4 |
| sunday | 1,000 | 3 |
| fly | 10,000 | 2 |
| under | 100,000 | 1 |
| the | 1,000,000 | 0 |

$$\text{idf}_t = \log_{10}(N/\text{df}_t)$$

There is one idf value for each term *t* in a collection.

Xi'an Jiaotong-Liverpool University 西交利物浦大学

## Exercise

Consider following documents with the stop word list: [when, in, the, and, I]

Doc 1: when walking in the rain

Doc 2: rain stopped walk, I ran, rain stop.

Doc 3: stop walking and run

Determine the document frequency and idf for the term **stop**.

Xi'an Jiaotong-Liverpool University 西交利物浦大学

## Effect of idf on ranking

- Does idf have an effect on ranking for one-term queries, like

  *iPhone?*

- Two terms queries:

  *iPhone price?*

37

## Effect of idf on ranking

- Does idf have an effect on ranking for one-term queries, like

  *iPhone?*
- idf has no effect on ranking one term queries

  idf affects the ranking of documents for queries with at least two terms

For the query capricious person, idf weighting makes occurrences of capricious count for much more in the final document ranking than occurrences of person.

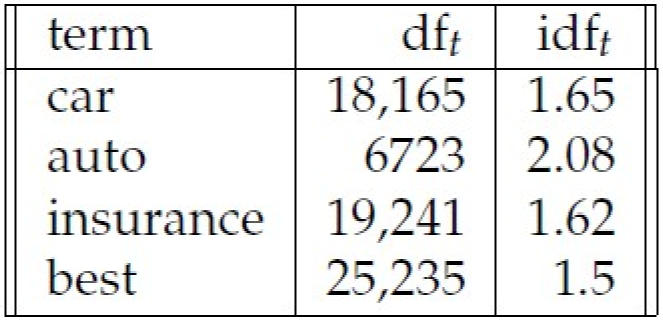
Xi'an Jiaotong-Liverpool University
西交利物浦大学

38

## idf weight

The idf of a rare term is high, whereas the idf of a frequent term is likely to be low.

idf is a measure of the informativeness of the term.



| term | $\mathrm{df}_t$ | $\mathrm{idf}_t$ |
|---|---:|---:|
| 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.

## Exercise

What is the idf of a term that occurs in every document?

## Solution

It is 0. For a word that occurs in every document, the word is ignored.