Programming for Social Scientists: Text Mining IV

黃瀚萱

Agenda

- Search
- Term frequency and Document Frequency
- Information Retrieval
  - TFIDF
  - BM25
- Indexing

Ranking

- There can be millions of documents matching the query terms.
    - It is impractical to show all results to human.
- Rank the documents according to their relevance to the query.

Term Frequency

- The occurrences of a term in a document.

$$tf(t,d)$$

Term Weighting

- Not all terms are equally important when it comes to assessing relevancy on a query.
- Certain terms have little or no discriminating value in determining relevance.
  - A collection of documents on the auto industry is likely to have the term auto in almost every document.
- Document frequency

$$df(t, D) = |\{d \in D : t \in d\}|$$

$$idf(t, D) = log \frac{N}{df(t, D)} = log \frac{|D|}{|\{d \in D : t \in d\}|}$$

D: The collection of documents.

$|D|$: Number of documents in the collection.

TF-IDF Model

- Combining the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document.

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

- Highest:
    - The term Frequently appears within a small number of documents.
- Middle:
    - The term appears fewer times in a document (lower TF), or appears in many documents (lower IDF).
- Lowest:
    - The term appears in almost all documents (Very low IDF).

TF-IDF Score

- For a query *q* containing a number of terms, the score of the query and a document *d* is measured with:

$$Score(q,d) = \sum_{t \in q} tfidf(t,d,D)$$

OKAPI BM25

- A modern IR model used in many applications.

$$score(q, d, D) = \sum_{t \in q} IDF(t, D) \times \frac{tf(t, d)(k_1 + 1)}{tf(t, d) + k_1(1 - b + b\frac{|d|}{avgdl})}$$

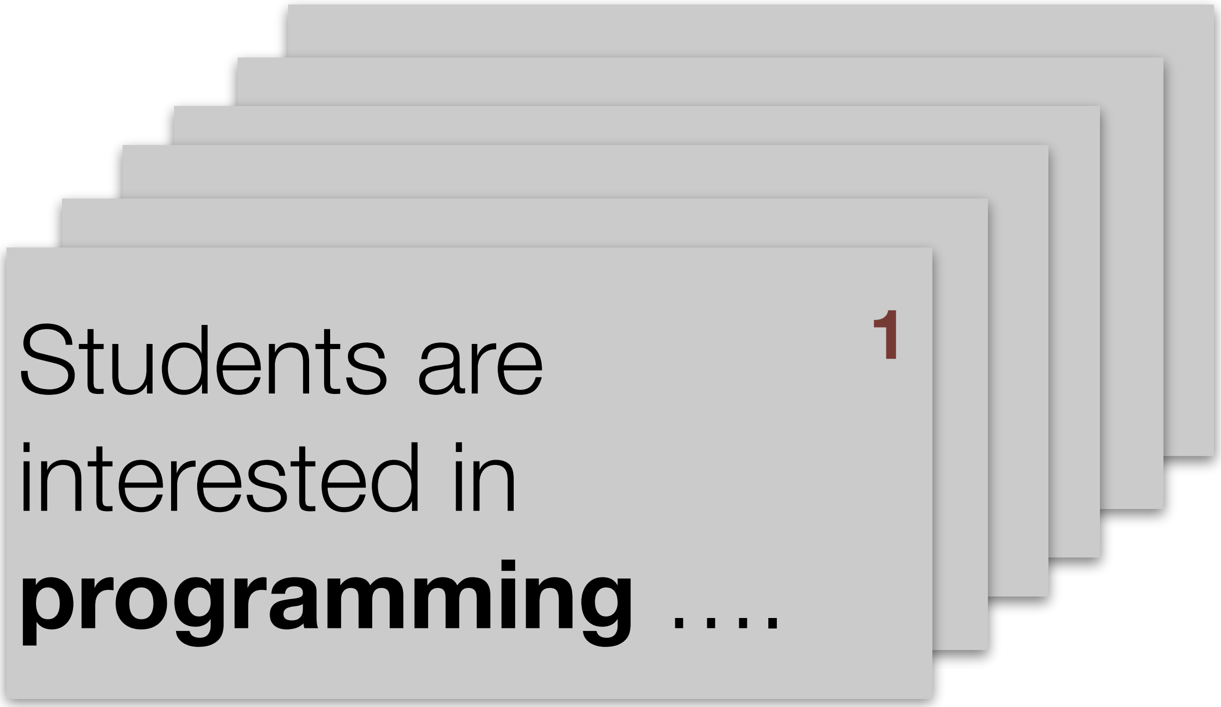
$$IDF(t) = log\frac{|D| - df(t, D) + 0.5}{df(t, D) + 0.5}$$

$$k_1 \in [1.2, 2.0], b = 0.75$$

Inverted Index

- Inverted indexing is the key to efficient search engine implementation.

Term	Documents
apple	1, 3, 5, 9, …
java	6, 7, 8, 12, …
interested	1, 3, 4, 14, …
social	22, 25, 71, …
justice	21, 28, 71, …
computation	3, 5, 8, 9, 11, …
programming	1, 3, 5, 6, 9, 12, …
students	1, 2, 4, 6, 8, 15, …



1

Students are interested in **programming** ….

Search with Inverted Index

- Build Inverted Index
    - Slow
    - Build once for each update
- Retrieval
    - Fetch postings list for each query term by looking up it in the inverted index.
    - Merge the postings lists.