Vector Space Model

Slides borrowed from Hongning Wang with modification

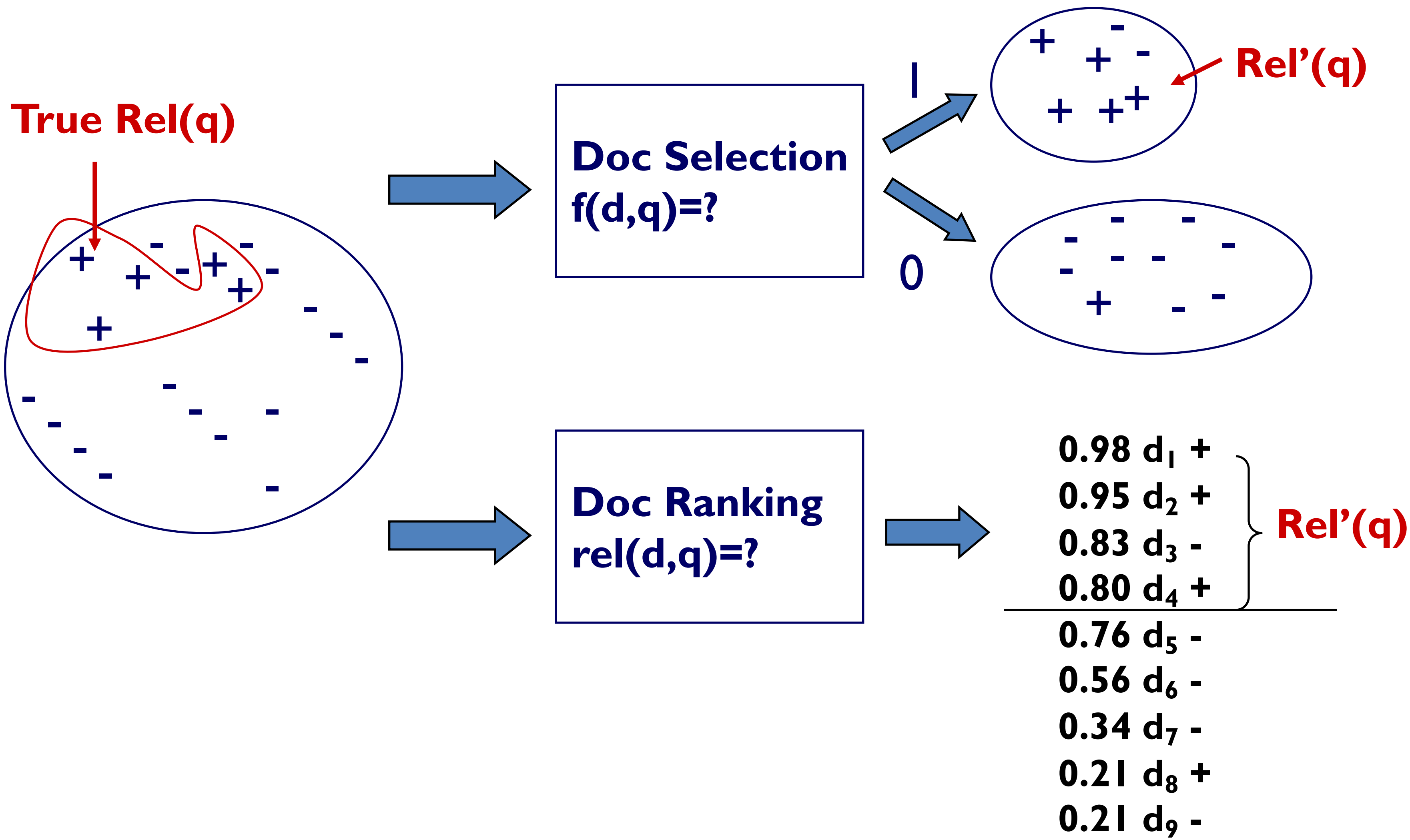
Ranked retrieval

- Thus far, our queries have all been Boolean
  - Documents either match or don't
- Can be good for expert users with precise understanding of their needs and the collection
  - Can also be good for applications: Applications can easily consume 1000s of results
- 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

Problem with Boolean search: feast or famine

- Boolean queries often result in either too few (=0) or too many (1000s) results.
- Query 1: "*standard user dlink 650*" → 200,000 hits
- Query 2: "*standard user dlink 650 no card found*": 0 hits
- It takes a lot of skill to come up with a query that produces a manageable number of hits.
  - AND gives too few; OR gives too many

Document Selection vs. Ranking



Intuitions for Ranking

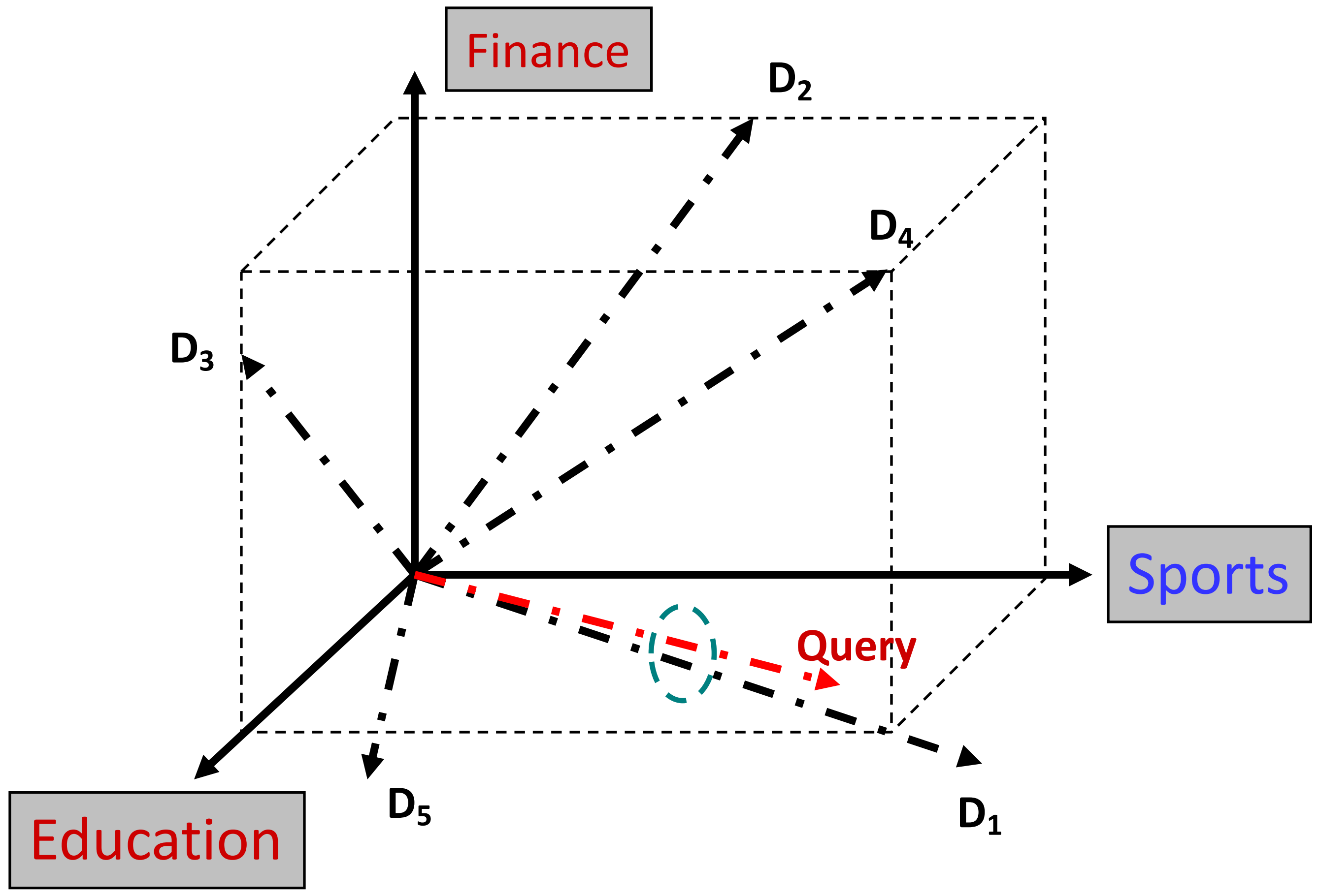
- Query side: some terms are more important than others to represent the user’s information need
- Document side: some terms carry more information about the document

Vector space model

- Represent both document and query by <u>concept</u> vectors
  - Each concept defines one dimension
  - *K* concepts define a high-dimensional space
  - Element of vector corresponds to concept weight
    - E.g., $d=(x_1,\ldots,x_k)$, x_i is "importance" of concept i
- Measure relevance
  - Similarity between the query vector and document vector in this concept space

VS Model: an illustration

- Which document is closer to the query?



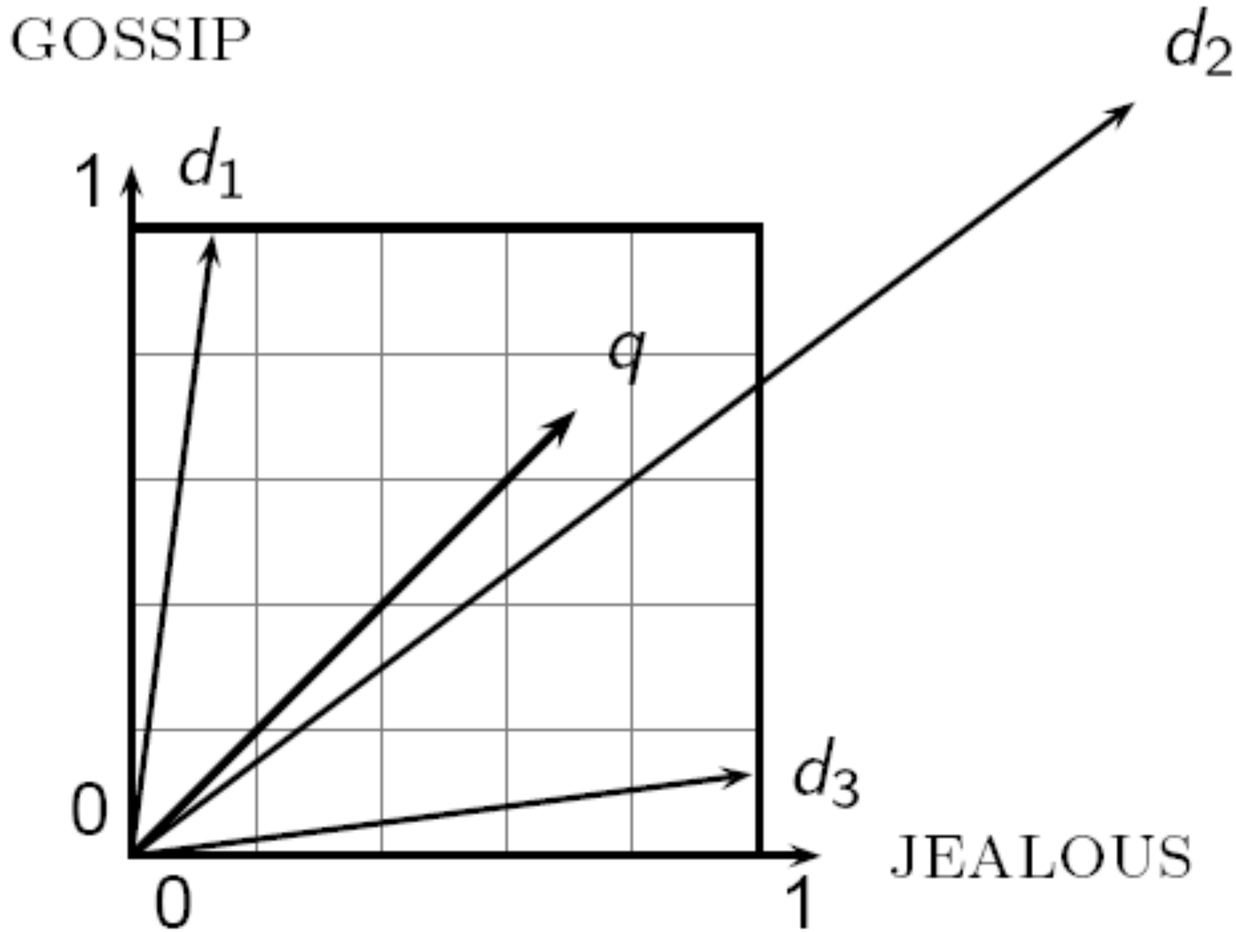
What the VS model doesn't say

- How to define/select the "basic concept"
  - Concepts are assumed to be orthogonal
- How to assign weights
  - Weight in query indicates importance of the concept
  - Weight in doc indicates how well the concept characterizes the doc
- How to define the similarity/distance measure

What is a good “basic concept”?

- Orthogonal
  - Linearly independent basis vectors
    - “Non-overlapping” in meaning
    - No ambiguity
- Weights can be assigned automatically and accurately
- Existing solutions
  - Terms or N-grams, i.e., bag-of-words
  - Topics, i.e., topic model

Bag of words representation



How to assign weights?

- Important!
- How?
  - Two basic heuristics
    - TF (Term Frequency) = Within-doc-frequency
    - IDF (Inverse Document Frequency)

TF weighting

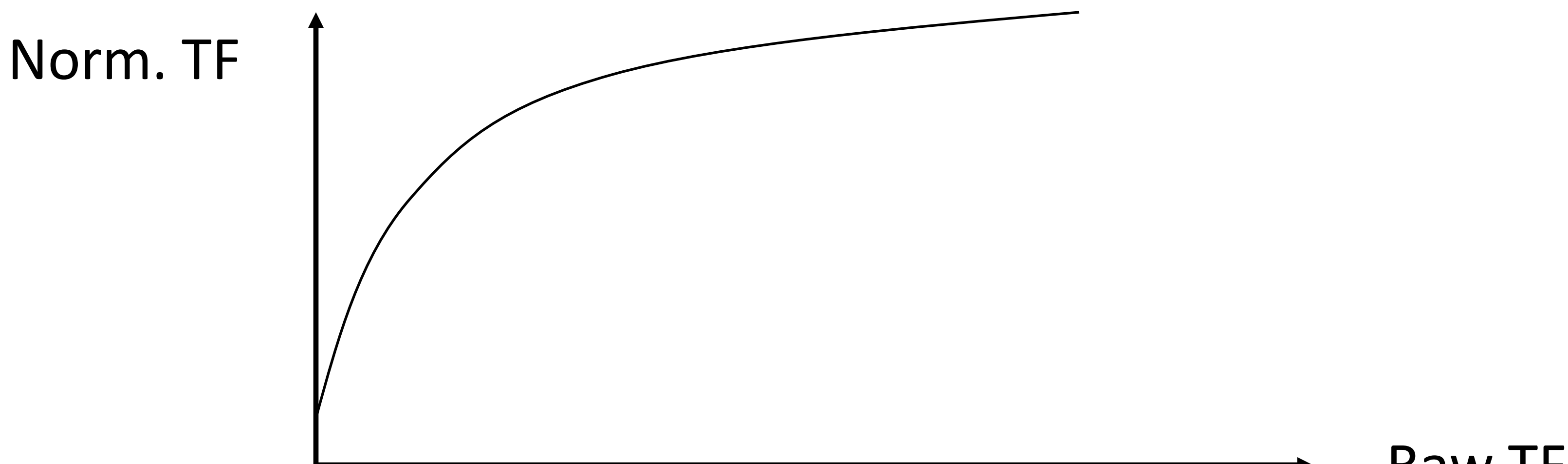
- Idea: a term is more important if it occurs more frequently in a document
- TF Formulas
  - Let $f(t,d)$ be the frequency count of term t in doc d
  - Raw TF: $tf(t,d) = f(t,d)$

TF normalization

- Query: *iphone 6s*
  - D1: iPhone 6s receives pre-orders on September 12.
  - D2: iPhone 6 has three color options.
  - D3: iPhone 6 has three color options. iPhone 6 has three color options. iPhone 6 has three color options.

TF normalization

- Sublinear TF scaling
  - $tf(t,d) = \begin{cases} 1 + \log f(t,d), if\ f(t,d) > 0 \\ 0, otherwise \end{cases}$



Document frequency

- Idea: a term is more discriminative if it occurs only in fewer documents

IDF weighting

- Solution
  - Assign higher weights to the rare terms
  - Formula
    - $IDF(t) = \log(\frac{N}{df(t)})$

Non-linear scaling

Total number of docs in collection

Number of docs containing term t

  - A corpus-specific property
    - Independent of a single document

Collection vs. Document frequency

- Collection frequency of *t* is the total number of occurrences of *t* in the collection (incl. multiples)
- Document frequency is number of docs *t* is in
- Example:

Word	Collection frequency	Document frequency
insurance	10440	3997
try	10422	8760

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

tf-idf weighting has many variants

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \ldots + w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N - \mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^\alpha$, $\alpha < 1$
L (log ave)	$\frac{1 + \log(\mathrm{tf}_{t,d})}{1 + \log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$				

TF-IDF weighting

- Combining TF and IDF
  - Common in doc → high tf → high weight
  - Rare in collection→ high idf→ high weight
  - $w(t, d) = TF(t, d) \times IDF(t)$
- Most well-known document representation schema in IR! (G Salton et al. 1983)

"Salton was perhaps the leading computer scientist working in the field of information retrieval during his time." - wikipedia

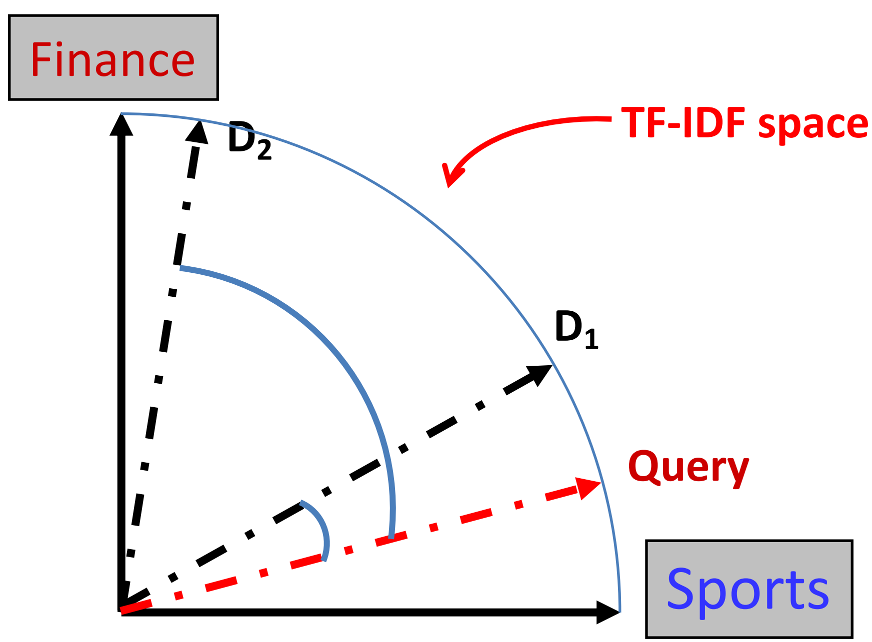
Gerard Salton Award – highest achievement award in IR

Cosine similarity

- Angle between two vectors
  - $cosine(V_q, V_d) = \frac{V_q \times V_d}{|V_q|_2 \times |V_d|_2} = \boxed{\frac{V_q}{|V_q|_2}} \times \frac{V_d}{|V_d|_2}$

    TF-IDF vector

    Unit vector
  - Document length normalized



What you should know

- Basic idea of vector space model
- Two important heuristics in VS model
  - TF
  - IDF
- Similarity measure for VS model
  - cosine similarity

Today's reading

- Chapter 6: Scoring, term weighting and the vector space model
  - 6.2 Term frequency and weighting
  - 6.3 The vector space model for scoring
  - 6.4 Variant tf-idf functions