Extending/Enhancing Model

1

Revisit: Information Retrieval

- User wants information from a collection of "objects": information need
- User formulates need as a "query"
  - Language of information retrieval system
- System finds objects that "satisfy" query
- System presents objects to user in "useful form"
- User determines which objects from among those presented are relevant

2

Revisit: Modeling

- Define each of the words in quotes
  - Information object
  - Query
  - Satisfying objects
  - Useful presentation: ranking

3

Models have seen

- Boolean model
  - Document: set of terms
  - Query: boolean expression over terms
  - Satisfying: evaluate boolean expression with respect to doc.
- Vector model
  - Dictionary of *t* terms
  - Document *t*-dimensional vector
    - Typically start as *bag* of terms
    - Specify how calculate weights of terms in docs.
  - Query *t*-dimensional vector
    - Typically start as set (maybe *bag)* of terms
    - Specify how calculate weights of terms in query
  - Satisfying:
    - Calculate a vector measure of similarity (document, query)
    - doc satisfies query if its score is >0
    - Documents are ranked by score

4

Start to enhance model

- Properties of terms within documents
  - Frequency of term in doc
  - Where in doc?
  - Special use? (e.g. in title, font, …)
  - Occurs in anchor text of another doc. pointing to this doc.

5

Start to enhance model

- Properties of terms within documents

Vector model gave us

  - Frequency of term in doc

Property of each occurrence of term in doc.

  - Where in doc?
  - Special use? (e.g. in title, font, …)

Found when evaluate *another* document

  - Occurs in anchor text of another doc. pointing to this doc.

6

Model

- Document: bag of terms + attributes
  - Basically sequence of terms. Why?
- Query: sequence of terms
  - Can make more complicated
- Satisfying: in current search engines, documents "containing" all terms
  - AND model
  - "containing" includes anchor text of pointers to this doc from other docs
- Ranking: wide open function of document and terms

7

Posting list

- For each document, keep list of terms appearing and attributes for each term
  - Actually list of positions at which each term occurs and attributes for that occurrence

Invert:

- For each term, keep list of documents in which it appears and attributes
  - For each document, list of positions at which term occurs and attributes for each occurrence

=> Inverted file/ Inverted index

8