Classic Information Retrieval

- Although search has changed, classic techniques still provide foundations – our starting point
- "classic" = foundational techniques for text documents without extra information about structure or content of a document

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

Information Retrieval cont.

- Define each of the words in quotes
  - Information object
  - Query
  - Satisfying objects
  - Useful presentation
- Notion of *relevance* critical
  - What really want?
  - Insufficient structure for exact retrieval
- Develop algorithms for the search and retrieval tasks

Think first about text documents

- Early digital searches – digital card catalog:
  - subject classifications, keywords
- "Full text" : words + English structure
  - No "meta-structure"
- Classic study
  - Gerald Salton SMART project 1960's

Scaling

- What are attributes changing from 1960's to online searches of today?
- How do they change problem?

Develop models

Begin with document models on board:

- Document is a ______ of terms*
  - Set
  - Bag
  - Sequence

* "term" is used instead of "word" to signal more general possibilities: serial numbers, nonsense, etc.

7

Modeling: "*query*"

Try

- *Query* is
  - Set of terms
  - Bag of terms
  - Sequence of terms
  - Other?
- What might query that is bag of terms mean?
- What might query that is sequence of terms mean?

8

Modeling: "*query*"

We will consider

- *Query*
  - Basic query is one term
  - Multi-term query is
    - Set of terms
    - sequence of terms
      - multiplicity?
      - Other constraints?
    - Boolean combination of terms

9

Modeling: "*satisfying*"

- What determines if document satisfies query?
- That depends ….
  - Document model
  - Query model
  - definition of "satisfying" can still vary
- *START SIMPLE*
  - *better understanding*
  - *Use components of simple model later*

10

AND Model

- Document: *set* of terms
- Query: *set* of terms
- Satisfying:
  - document satisfies query if all terms of query appear in document

Currently used by Web search engines

11

OR Model

- Document: *set* of terms
- Query: *set* of terms
- Satisfying:
  - document satisfies query if one or more terms of query appear in document

Original IR model

why?

12

(pure) Boolean Model of IR

- Document: *set* of terms
- Query: Boolean expression over terms
- Satisfying:
  - Doc. evaluates to "true" on single-term query if contains term
  - Evaluate doc. on expression query as you would any Boolean expression
  - doc satisfies query if evals to true on query

13

Boolean Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query:

(principles AND knowledge) OR (science AND engineering)

0 1 1 0

Doc 1: FALSE

14

Boolean Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query:

(principles AND knowledge) OR (science AND engineering)

1 0 1 1

Doc 2: TRUE

15

Boolean Model example 2

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query: (principles OR knowledge) AND (science AND NOT(engineering))

Doc 1: (0 OR 1) AND (1 AND NOT(0)) TRUE

16

Boolean Model example 2

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query: (principles OR knowledge) AND (science AND NOT(engineering))

Doc 2: (1 OR 0) AND (1 AND NOT(1)) FALSE

17

(pure) Boolean Model of IR: how "*present results in useful form*"

- can refer to user interface
- more basic: give list of results
- meaning of order of list? => RANKING?

- There is no ranking algorithm in pure Boolean model
  - Ideas for making one without changing semantics of "satisfy"?

18

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query:
(principles OR knowledge) AND (science OR engineering)

Doc 1:	0	1	1	0	TRUE
Doc 2:	1	0	1	1	TRUE

RANK?

19

Introducing Ranking

- Order documents that satisfy a query by how well match the query
- How capture relevance to user by algorithmic method of ordering?

20

Simplified Vector Model

- Document: *bag* of terms - count occurrences
- Query: *set* of terms
- Satisfying: OR model
- Ranking: numerical score measuring degree to which document satisfies query

  some choices:
  - one point for each query term in document
  - ✓one point for ***each occurrence*** of a query term in document
- Documents returned in sorted list by decreasing score

21

Simplified Vector Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Frequencies:
science 1; knowledge 2; principles 0; engineering 0

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Frequencies:
science 2; knowledge 0; principles 1; engineering 1

22

Ranking

- What intuitive criteria?

23

Enhanced document model

- First model: set of terms
  - term in/not in document
- Next: bag of terms
  - know frequency of terms in document
- Now: sequence of terms + additional properties of terms
  - sequence gives you where term in doc
    - derive relative position of multiple query terms
  - Special use? (e.g. in title, font, …)
    - most require "mark-up": tags, meta-data, etc.

24

HTML mark-up example

<h2> Communication </h2>
This course will be essentially ``paperless''. All assignments will be posted <i>only</i> on the course Web site. ``Handouts'' and copies of any transparencies used in class will be posted on the course Web site as well. Important announcements on all aspects of the course will be made on the Announcements page. <b>Students are responsible for monitoring the postings under ``Announcements''. </b> Schedule changes will be made on the on-line <a href="schedule.html"> schedule page</a>. and announced under ``Announcements''. The only paper we will exchange is your solutions to the problem sets, which we will grade and hand back, the exam questions and your responses, and your project reports.

25

yields

Communication

This course will be essentially ``paperless''. All assignments will be posted *only* on the course Web site (see Schedule and Readings). ``Handouts'' and copies of any transparencies used in class will be posted on the course Web site as well. Important announcements on all aspects of the course will be made on the Announcements page. **Students are responsible for monitoring the postings under ``Announcements''.** Schedule changes will be made on the on-line schedule page. and announced under ``Announcements''. The only paper we will exchange is your solutions to the problem sets, which we will grade and hand back, the exam questions and your responses, and your project reports.

26

Enhanced document model: restate

"sequence of terms + properties of terms"

⇓ WHY?

"set of (term, properties) pairs"

Properties:
- for each distinct term
  - Frequency of term in doc
    - Vector model of classic IR
- for individual occurrence of each term
  - Where in doc.
  - properties of use

27

Model

- Document: set of (term,properties) pairs
- Query: sequence of terms
  - Can make more complicated
- Satisfying: AND model
  - relax if no document contains all?
- Ranking: wide open function
  - info beyond documents and query ?

28

Data Structure for Collection

- for each document, keep list of:
  - terms appearing
    - aggregate properties of term
      e.g. frequency
    - positions at which each term occurs
      - attributes for each occurrence of term
- keep summary information for documents

29

Data Structure for Collection: Invert

- for each term, keep list of:
  - documents in which it appears
    - positions at which it occurs in each doc.
      - attributes for each occurrence
- keep summary information for documents
- keep summary information for terms

30

Inverted Index for Collection

- for each term, keep POSTINGS LIST of:
  - each document in which it appears
    - each position at which it occurs in doc.
      - attributes for each occurrence

  POSTING

- Core structure used by query evaluation and document ranking algorithms

31

Index structure

$term_1$:(doc ID (position, attributes)
(position, attributes),
…
(position, attributes))
(doc ID (position, attributes)
(position, attributes),
…
(position, attributes))
…
$term_2$:(doc ID (position, attributes)
(position, attributes),
…
(position, attributes))
…

32

Classic IR Ranking: the Vector Model

33

Vector Model

- Document: *bag* of terms
- Query: list of terms (imprecise)
- Satisfying:
  - Each document is scored as to the degree it satisfies query (non-negative real number)
  - doc satisfies query if score is > threshold
  - Documents are returned in sorted list decreasing by score:
    - Include only scores above threshold
    - Include only highest *n* documents, some *n*

34

How compute score?

1. There is a *dictionary* (aka *lexicon*) of all terms, numbering *t* in all
   - Number the terms 1, …, *t*
2. Change the model of a document (temporarily):
   - A document is a *t*-dimensional vector
   - The i^{th} entry of the vector is the *weight* (importance of) term *i* in the document
3. Change the model of a query (temporarily):
   - A query is a *t*-dimensional vector
   - The i^{th} entry of the vector is the *weight* (importance of) term *i* in the query

35

How compute score, continued

4. Calculate a vector function of the document vector and the query vector to get the score of the document with respect to the query.

Choices:

1. Measure the distance between the vectors:

   $\text{Dist}(\boldsymbol{d},\boldsymbol{q}) = \sqrt{(\Sigma^t_{i=1}(\boldsymbol{d}_i - \boldsymbol{q}_i)^2)}$

   - Is *dissimilarity* measure
   - Not normalized: Dist ranges [0, inf.)
   - Fix: use $e^{-\textbf{Dist}}$ with range (0,1]
   - Is it the right sense of difference?

36

How compute score, continued

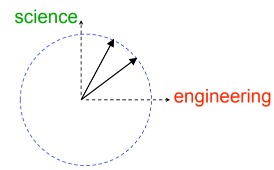
2. Measure the angle between the vectors:
   Dot product: $\boldsymbol{d} \cdot \boldsymbol{q} = \Sigma^t_{i=1}(\boldsymbol{d}_i * \boldsymbol{q}_i)$

- Is *similarity* measure
- Not normalized: dot product ranges (-inf., inf.)
- Fix: use normalized dot product, range [-1,1]
  $(\boldsymbol{d} \cdot \boldsymbol{q}) / (|\boldsymbol{d}| * |\boldsymbol{q}|)$ where $|\boldsymbol{v}| = \sqrt{\Sigma^t_{i=1}(\boldsymbol{v}_i^2)}$
  the length of ***v***
- In practice vector components are non-negative so range is [0,1]
- This most commonly used function for score

37

Normalizing vectors

- If use unit vectors, $\boldsymbol{d} / |\boldsymbol{d}|$ and $\boldsymbol{v} / |\boldsymbol{v}|$ some of issues go away



38

How compute weights d_i and q_i?

First:

observations about this model?

39

Vector model: Observations

- Have matrix of terms by documents
  ⇒Can use linear algebra
- Queries and documents are the same
  ⇒Can compare documents same way
  - Clustering documents
- Document with only some of query terms can score higher than document with all query terms

40

How compute weights

- Vector model *could* have weights assigned by human intervention
  - may add meta-information
  - User setting query weights might make sense
    - User decides importance of terms in own search
  - Humans setting document weights?
    - Who? Billions+ of documents
- Return to model of documents as bag of words – calculate weights
  - Function mapping bag of words to vector

41

Summary weight calculation

- General notation:
  - w_{jd} is the weight of term *j* in document *d*
  - $freq_{jd}$ is the # of times term *j* appears in doc *d*
  - n_j = # docs containing term j
  - *N* = number of docs in collection
- Classic *tf-idf* definition of weight:
  $w_{jd} = freq_{jd} * \log(N/n_j)$

42

Summary weight calculation

- General notation:
  - w_{jd} is the weight of term *j* in document *d*
  - $freq_{jd}$ is the # of times term *j* appears in doc *d*
  - n_j = # docs containing term j
  - *N* = number of docs in collection
- Classic *tf-idf* definition of weight, normalized:

$$u_{jd} = freq_{jd} * \log(N/n_j)$$

$$w_{jd} = \frac{u_{jd}}{(\Sigma^t_{i=1}(u_{id}^2))^{1/2}}$$

43

Weight of query components?

- Set of terms, ***some choices***:
  1. w_{jq} = 0 or 1
  2. $w_{jq} = freq_{jq} * \log(N/n_j)$ $= 0 \text{ or } \log(N/n_j)$
- Bag of terms
  - Analyze like document

    Some queries are prose expressions of ***information need***

Do we want idf term in both document weights and query weights?

44

Vector Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Frequencies:

science 1; knowledge 2; principles 0; engineering 0

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Frequencies:

science 2; knowledge 0; principles 1; engineering 1

45

Vector model example cont.

- Consider the 5 100-level and 200-level COS courses as the collection (109, 217, 226)
- Only other appearance of our 4 words is "science" once in 109 description.
- idf:

  science ln(5/3) = .51

  engineering, principles, knowledge:

  ln(5/1) = 1.6

46

Term by Doc. Table: $freq_{jd} * \log(N/n_j)$

	Doc 1	Doc 2
science	.51	1.02
engineering		1.6
principles		1.6
knowledge	3.2	

47

Unnormalized dot product for query:

science, engineering, knowledge, principles

using 0/1 query vector

- Doc 1: 3.71
- Doc 2: 4.22
- If documents have about same vector length, this right ratio for normalized (cosine) score

48

Additional ways to calculate document weights

- Dampen frequency effect:
  w_{jd} = 1+ log ($freq_{jd}$) if $freq_{jd}$ > 0; 0 otherwise
- *Use smoothing term to dampen effect:*
  W_{jd} = a + (1-a) $freq_{jd}$ / max_p ($freq_{pd}$)
  - *a is typically .4 or .5*
  - *Can multiply second term by idf*
- *Effects for long documents (Section 6.4.4)*

49

Where get dictionary of *t* terms?

- Pre-determined dictionary.
  - How sure get all terms?
- Build lexicon when collect documents
  - What if collection dynamic: add terms?

50

Classic IR models - Taxonomy

Well-specified models:
(not extended model)

✓Boolean
✓Vector

- Probabilistic
  - based on probabilistic model of words in documents

51

Models have seen

Model	Document	Query	Satisfy
Boolean	set of terms	Boolean expression over terms	evaluate boolean expression
Vector dictionary of *t* terms	*t*-dimensional vector	*t*-dimensional vector	vector measure of similarity Doc.s ranked by score
Extended	set of pairs (term, properties)	sequence of terms	Boolean AND Doc.s ranked; flexible scoring algorithm

52

Query models advantages

- Boolean
  - \- No ranking in pure
  - \+ Get power of Boolean Algebra:
    - expressiveness
    - optimization of query forms

versus

- Vector
  - \+ Query and document look the same
  - \+ Power of linear algebra
  - \- No requirement all terms present in pure

53

Query models advantages

- Extended
  - \+ could use full Boolean Algebra to define satisfying documents
  - \- query and document not same
  - • ranking arbitrary function of document and query
    - \- computational cost?

54