Information Retrieval Overview

Vagelis Hristidis
School of Computer Science
Florida International University
COP 6727

Roadmap

- What is IR?
- Matching Models
- Evaluation of Results
- Digital Libraries vs. IR
- Bridging IR + Databases
- Proximity Search in Databases [Goldman et al.]

What IR Systems Try to Do

- Predict, on the basis of some information about the user, and information about the knowledge resource, what information objects are likely to be the most appropriate for the user to interact with, at any particular time

How IR Systems Try to Do This

- Represent the user's information problem (the *query*)
- Represent (*surrogate*) and organize (*classify*) the contents of the knowledge resource
- Compare query to surrogates (predict relevance)
- Present results to the user for interaction/judgment

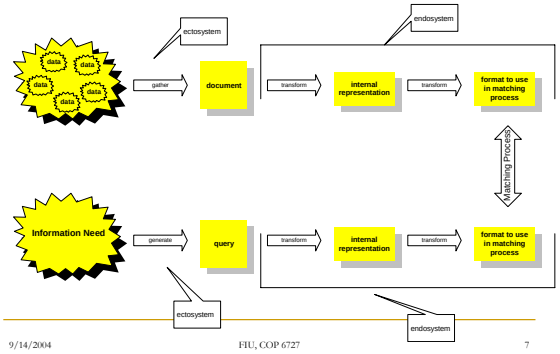
Why IR is Difficult

- People cannot specify what they don't know (Anomalous State of Knowledge), so representation of information problem is inherently uncertain
- Information objects can be about many things, so representation of aboutness is inherently incomplete

Document & Query

- Document Side
  - generate data ➧ document
  - transform ➧ internal representation ➧ match
- Query Side
  - information need ➧ generate query
  - transform ➧ internal representation ➧ match
- Various structures that have been proposed and used for queries to a retrieval systems

The document and the query undergo parallel processes within the retrieval system.



Roadmap

- What is IR?
- Matching Models
- Evaluation of Results
- Digital Libraries vs. IR
- Bridging IR + Databases
- Proximity Search in Databases [Goldman et al.]

Matching Criteria

- An exact match can only be found in special situations
  - requires precise query
  - numerical or business applications
- Range Match
  - works best in a DB with defined fields
- Approximate Matching
- Matching techniques can be combined
  - eg, begin with approximate, narrow down with exact or range

Boolean Queries

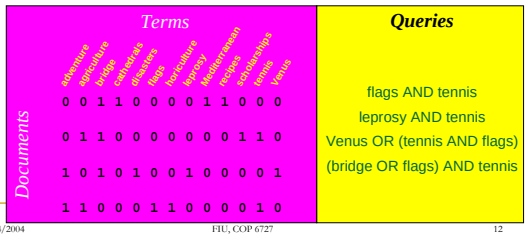
- Based on concepts from logic: AND, OR, NOT
- Order of operations (two conventions)
  - NOT, AND, OR
  - left to right
- Standard forms
  - Disjunctive Normal Form (DNF)
    - Terms, Conjuncts, Disjuncts
    - (P AND Q) OR (Q AND NOT R) OR (P AND R)
  - Conjunctive Normal Form (CNF)
    - Terms, Disjuncts , Conjuncts
    - P AND (NOT Q OR R) AND (S OR NOT R)

Truth Table

P	Q	NOT P	P AND Q	P OR Q
0	0	TRUE	FALSE	FALSE
0	1	TRUE	FALSE	TRUE
1	0	FALSE	FALSE	TRUE
1	1	FALSE	TRUE	TRUE

Boolean-based Matching

- Separate the documents containing a given term from those that do not.
- No similarity between document and query structure
- Proximity Judgement : Gradations of the retrieved set



Exact Match IR

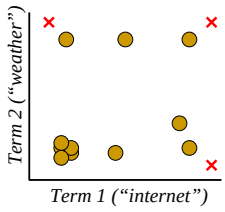
- Advantages
  - Efficient
  - Boolean queries capture some aspects of information problem structure
- Disadvantages
  - Not effective
  - Difficult to write effective queries
  - No inherent document ranking

Vector Queries

- Documents and Queries are vectors of terms
- Actual vectors have many terms (thousands)
- Vectors can be Boolean (keyword) or weighted (term frequencies)
- Example terms: "dog","cat","house", "sink", "road", "car"
- Boolean: (1,1,0,0,0,0), (0,0,1,1,0,0)
- Weighted: (0.01,0.01, 0.002, 0.0,0.0,0.0)
- Queries can be weighted also*

Vector-based Matching: Metrics

- Metric or Distance Measure : document close together in the vector space are likely to be highly similar



Axes represent terms

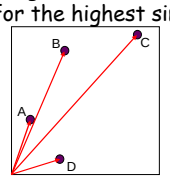
Documents are represented as points in the vector space

Similarities among documents

Similarities between documents and queries

Vector-based Matching: Cosine

- Cosine of the angle between the vectors representing the document and the query
- Documents "in the same direction" are closely related.
- Transforms the angular measure into a measure ranging from 1 for the highest similarity to 0 for the lowest



Example, continued

- Document A: "A dog and a cat."
  - Vector: (2,1,1,1,0)

a	and	cat	dog	frog
2	1	1	1	0

- Document B: "A frog."
  - Vector: (1,0,0,0,1)

a	and	cat	dog	frog
1	0	0	0	1

Queries

- Queries can be represented as vectors in the same way as documents:
  - Dog = (0,0,0,1,0)
  - Frog = ( )
  - Dog and frog = ( )

Similarity measures

- There are many different ways to measure how similar two documents are, or how similar a document is to a query
- The cosine measure is a very common similarity measure
- Using a similarity measure, a set of documents can be compared to a query and the most similar document returned

The cosine measure

- For two vectors d and d' the cosine similarity between d and d' is given by:

$$\frac{d \times d'}{|d||d'|}$$

- Here d X d' is the vector product of d and d', calculated by multiplying corresponding frequencies together
- The cosine measure calculates the angle between the vectors in a high-dimensional virtual space

Example

- Let d = (2,1,1,1,0) and d' = (0,0,0,1,0)
  - dXd' = 2X0 + 1X0 + 1X0 + 1X1 + 0X0=1
  - |d| = $\sqrt{(2^2+1^2+1^2+1^2+0^2)} = \sqrt{7}=2.646$
  - |d'| = $\sqrt{(0^2+0^2+0^2+1^2+0^2)} = \sqrt{1}=1$
  - Similarity = 1/(1 X 2.646) = 0.378
- Let d = (1,0,0,0,1) and d' = (0,0,0,1,0)
  - Similarity = 0

Vector Space Model

- Advantages
  - Straightforward ranking
  - Simple query formulation (bag of words)
  - Intuitively appealing
  - Effective
- Disadvantages
  - Unstructured queries
  - Effective calculations and parameters must be empirically determined

Fuzzy Queries

- Fuzzy Logic: Propositions have a "truth value" between 0 and 1
- Fuzzy NOT: $1-t$
- Fuzzy AND: $t_1 * t_2$
- Fuzzy OR: $1-(1-t_1)*(1-t_2)$
- Example:
  - All swans are white 0.8 1
  - All swans can swim 0.9 1
  - White and swim 0.72 1
  - White or swim 0.92 1

Probabilistic Queries

- Like fuzzy queries, except they adhere to the laws of probability
- Can use probabilistic concepts like Baye's Theorem
- Term frequency data can be used to estimate probabilities

Natural Language Queries

- The “Holy Grail” of information retrieval
- Issues in Natural Language Processing
  - syntax
  - semantics
  - pragmatics
  - speech understanding
  - speech generation

Vocabulary

- Stopword lists
  - Commonly occurring words are unlikely to give useful information and may be removed from the vocabulary to speed processing
  - Stopword lists contain frequent words to be excluded
  - Stopword lists need to be used carefully
    - E.g. “to be or not to be”

Term weighting

- Not all words are equally useful
- A word is most likely to be highly relevant to document A if it is:
  - Infrequent in other documents
  - Frequent in document A
  - A is short
- The cosine measure needs to be modified to reflect this

Normalised term frequency (tf)

- A normalised measure of the importance of a word to a document is its frequency, divided by the maximum frequency of any term in the document
- This is known as the tf factor.
- Document A: raw frequency vector: (2,1,1,1,0), tf vector: ( )
- This stops large documents from scoring higher

Inverse document frequency (idf)

- A calculation designed to make rare words more important than common words
- The idf of word *i* is given by

$$idf_i = \log \frac{N}{n_i}$$

- Where *N* is the number of documents and n_i is the number that contain word *i*

tf-idf

- The tf-idf weighting scheme is to multiply each word in each document by its tf factor and idf factor
- Different schemes are usually used for query vectors
- Different variants of tf-idf are also used

Document Scoring Functions

tf	is the term's frequency in document
qtf	is the term's frequency in query
N	is the total number of documents in the collection
df	is the number of documents that contain the term
dl	is the document length (in bytes), and
$avdl$	is the average document length

Okapi weighting based document score: [23]

$$\sum_{t \in Q,D} ln\frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1((1 - b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

k_1 (between 1.0–2.0), b (usually 0.75), and k_3 (between 0–1000) are constants.

Pivoted normalization weighting based document score: [30]

$$\sum_{t \in Q,D} \frac{1 + ln(1 + ln(tf))}{(1 - s) + s\frac{dl}{avdl}} \cdot qtf \cdot ln\frac{N + 1}{df}$$

s is a constant (usually 0.20).

Missing Terms & Term Relationships

- Vector space model problem
- 0 value is used in 2 ways :
  - indicate terms that are truly missing
  - indicate terms about which there is no information
- Problem : relationships among the terms in a document or query.
- Linearly independent set of basis vector

Probabilistic Matching

- Given a document and a query it should be possible to calculate the probability that the document is relevant to the query.
- Assumption : the number of document within the database that are relevant to the query is known.
- Discriminant function (dis(selected) > 1)
- Much calculation and many assumption
- Good results, but not better than those obtained using Boolean or vector model

Probabilistic IR

- Advantages
  - Straightforward relevance ranking
  - Simple query formulation
  - Sound mathematical/theoretical model
  - Effective
- Disadvantages
  - Unrealistic assumptions (term independence)
  - Probabilities difficult to estimate

Fuzzy Matching

- Replaces the need to estimate probabilities by a need to estimate a sense of belief about a document relevant.
- Fuzzy matching :
  - calculation based on defined membership grades for terms
  - how well a related term matches a given term.
  - Modifiers or descriptors
- Problem : how such terms translate into the membership function associated with fuzzy retrieval

Proximity Matching

- Proximity Criteria
- Additional criteria to further refine the set of document identified by one of the other matching methods.
- Modification of proximity criteria
  - use phrases rather than simple word proximity
  - ordered proximity to aid in the retrieval decision

Roadmap

- What is IR?
- Matching Models
- Evaluation of Results
- Digital Libraries vs. IR
- Bridging IR + Databases
- Proximity Search in Databases [Goldman et al.]

Evaluation of IR Systems

- Traditional goal of IR is to retrieve *all* and *only* the relevant documents in response to a query
- All is measured by *recall*: the proportion of relevant documents in the collection which are retrieved
- Only is measured by *precision*: the proportion of retrieved documents which are relevant

Evaluation Problems

- Realistic IR is interactive; traditional IR methods and measures are based on non-interactive situations
- Evaluating interactive IR requires human subjects; the normal mode of evaluation is comparison between two systems (no gold standard or benchmarks); cannot compare a subject's searching on the same task in two systems

How Interaction Has Been Accounted For

- Relevance feedback
  - Automatically moving the initial query toward the "ideal" query
  - Term reweighting and query expansion
- Support for query modification
  - Display of "good" and "bad" terms
  - Thesauri
  - Inter-document relations

Roadmap

- What is IR?
- Matching Models
- Evaluation of Results
- Digital Libraries vs. IR
- Bridging IR + Databases
- Proximity Search in Databases [Goldman et al.]

What is a Digital Library (DL)?

- "a collection of information that is both digitized and organized" (Lesk, p. 1)
  - there are any number of alternate definitions, but this seems fair enough
  - no mention of architecture, implementation, content, etc.

How is a DL different from a database?

- A traditional SQL database has as its basic element data items in a relation:

```
select name
from employee, project
where employee.deptnumber = "25" AND
      project.number = "100"
```

- databases exploit known structures and relations
- DBMS retrieval is not probabilistic (Frakes, Baeza-Yates, p. 3)

How is a DL different from traditional IR systems?

- The difference is less clear
- IR systems can be considered the precursors to DLs
- The basic unit of a IR system is a document and the focus is on *textual retrieval*
  - exact matching - Boolean, text pattern searching
  - inexact matching - probabilistic, vector space, clustering

How is a DL different from the WWW?

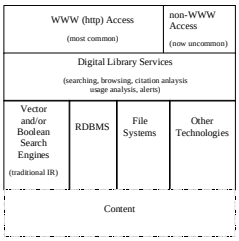
- The key difference is *organization*
  - The WWW as a whole has no real organization
- Recently, convergence as search engines (Google) attempt to add an organizational framework to their web holdings
  - In the past, most are focused on keyword searching (i.e., Altavista)

How is a DL different from the WWW?

- Another key difference is who controls the input into the system
  - most meta searchers hunt down their holdings
    - Lycos is short for *Lycosidae lycosa* (the "wolf spider"), which pursues its prey and does not build a web (Mauldin, IEEE Expert, 1/97)
  - some (Yahoo) have humans in the loop for review and classification
- To date, DLs are generally more tightly controlled, and have a targeted customer set

DL = Content + Services

digital library = collection of information both digitized and organized
-- M. Lesk, 1997



- DL is the union of the content and services defined on the content

Roadmap

- What is IR?
- Matching Models
- Evaluation of Results
- Digital Libraries vs. IR
- Bridging IR + Databases
- Proximity Search in Databases [Goldman et al.]

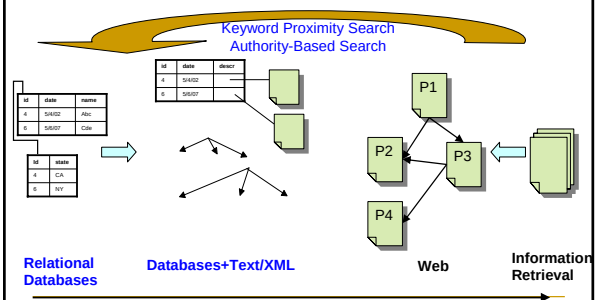
Database Search vs. Information Retrieval (Document Search)

Database Search	*Information Retrieval*
Data Stored in Structured Data Types (Tables, XML) and Conform to Schema	Data is Set of Documents
d Query (Query)	pressions
{o51, o24}	'Smith AN Doc1 0.9 Doc3 0.7 Doc2 0.5
SELECT orderid FROM customers C, orders O, suppliers S WHERE C.custname="Miller" AND S.supplname="Smith" AND C.custid=O.custid AND O.suppid=S.suppid	Answer: Ranked List of Documents

VH4 VH5

VH6

Bridging the Gap Between Databases and Information Retrieval



VH7

Goal

- Currently, Information Discovery in Databases Requires:
  - Knowledge of the Role of the Keywords
  - Knowledge of Schema
  - Knowledge of a Query Language
- Enable IR Without t

1. SELECT * FROM customers,orders,s… custname="Miller" AND supplname="Smith" AND …
2. SELECT * FROM customers,orders,suppliers WHERE custname="Smith" AND supplname="Miller" AND …
3. …

"Miller Smith"

Query Type

Structured Queries (SQL, XQuery) | Semistructured Queries | Keyword Search

VH8

Database Graph

- Application Specific
- Node can be attribute value, tuple, group of tuples…
- Edges are semantic connections
  - Primary to foreign keys
  - XML edges
  - …

Roadmap

- What is IR?
- Matching Models
- Evaluation of Results
- Digital Libraries vs. IR
- Bridging IR + Databases
- Proximity Search in Databases [Goldman et al.]

Proximity Search in Databases [VLDB1998]

- Find Set
- Near Set
- E.g., Find Movie near Travolta Cage
  - Near set = {Travolta, Cage}
  - Find set = ?

Slide 49

VH4 which require that we specify the types of connections between keywords
Vagelis, 2/14/2004

VH5 SQL has logical underpinnings, which require that each tuple is either in or out of the result.
Vagelis, 2/13/2004

Slide 50

VH6 Databases and IR have followed distinct research ways, since they were considered fundamentally different.

However in the last years, we have witnessed a convergence between the two areas.

From the rigidly structured relational databases with atomic attributes, we moved to databases with plain text attributes and to semistructured data like XML which combine plain text with structured elements.

On the other end from IR we moved to Web Search, where there are links between the documents (pages).

However, the information discovery techniques have remained seperated.

In an effort to bridge this gap, we carry two well-studied ideas of IR and Web search to database. In particular, keyword proximity search discovers connections between keywords, and authority-based searc adapts the idea of PageRank to databases.
Vagelis, 2/22/2004

Slide 51

VH7 Let's see which are the requirements for searching a db.

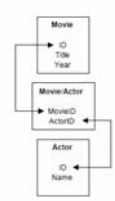
Suppose we want to discover how Smith and Miller are associated.
To do so using traditional discovery techniques we need to know the role of the keywords (if Smith is a customer or supplier etc), the schema (eg: customers are connected with the suppliers through the orders relation), and a query language (SQL).

If we don't know how the keywords are connected, we would have to write a large number of SQL queries like...

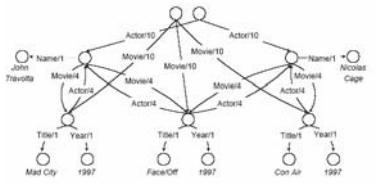
Our works enables ....
Vagelis, 2/22/2004

VH8 structured and keyword queries are the two extremes of the query type axis. In the middle lie semistructured queries which are part of my future work and will not be discussed in this presentation.
Vagelis, 2/22/2004

Example Movie Database



Example (cont'd)



Example (cont'd)

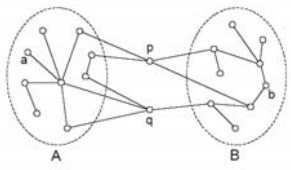
Ranking Function

- Bond between nodes f, n
  $b(f,n) = r_F(f) r_N(n) / d(f,n)^t$
  - r_N, r_F : ranking in set N, F
  - d: distance
- $Score(f) = Sum_{n \in N} (b(f,n))$
- Or max(…)

Execution

- Dijkstra
  - Efficient for in memory only
- Precompute all paths (Floyd-Warshall)
  - Inefficient in time and space
  - No way to prune for distance>K
- Present algorithm to compute all-pairs distances efficient for graphs stored on disks
  - Still too much space!

Hub indexing



- Hub index consists of:
  - Hub set H and shortest distances between them
  - Distances between pairs of objects not crossing through H
- Algorithm to efficiently answer query using the hub index
- Hub set is nodes with highest degree (heuristic)

Some References

- Some slides have been taken from:
  - Michael L. Nelson, ODU
  - Paul Minro
  - Nicholas J. Belkin, Rutgers
  - Peter Burden
- Goldman et al. Proximity Search in Databases. VLDB 1998