LBSC 796/INFM 718R: Week 3

Boolean and Vector Space Models

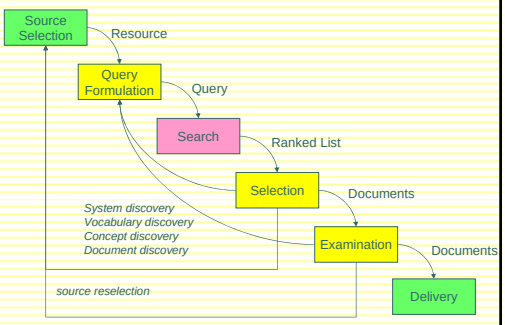
Jimmy Lin
College of Information Studies
University of Maryland

Monday, February 13, 2006

Muddy Points

- Statistics, significance tests
- Precision-recall curve, interpolation
- MAP
- Math, math, and more math!
- Reading the book

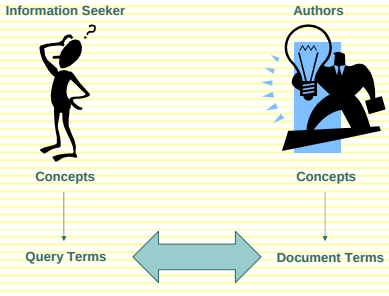
The Information Retrieval Cycle



What is a model?

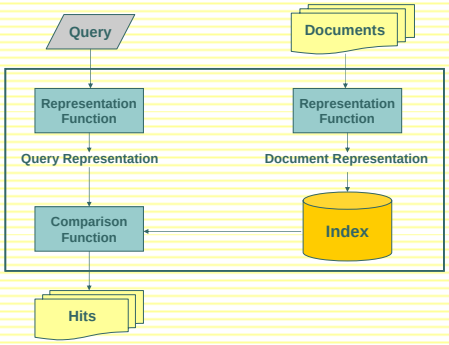
- A model is a construct designed help us understand a complex system
  - A particular way of "looking at things"
- Models inevitably make simplifying assumptions
  - What are the limitations of the model?
- Different types of models:
  - Conceptual models
  - Physical analog models
  - Mathematical models
  - …

The Central Problem in IR



Do these represent the same concepts?

The IR Black Box



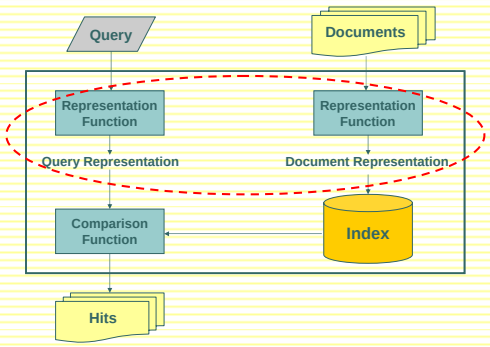
Today's Topics

- Boolean model
  - Based on the notion of sets
  - Documents are retrieved *only* if they satisfy Boolean conditions specified in the query
  - Does not impose a ranking on retrieved documents
  - Exact match
- Vector space model
  - Based on geometry, the notion of vectors in high dimensional space
  - Documents are ranked based on their similarity to the query (ranked retrieval)
  - Best/partial match

Next Time…

- Language models
  - Based on the notion of probabilities and processes for generating text
  - Documents are ranked based on the probability that they generated the query
  - Best/partial match

Representing Text



How do we represent text?

- How do we represent the complexities of language?
  - Keeping in mind that computers don't "understand" documents or queries
- Simple, yet effective approach: "bag of words"
  - Treat all the words in a document as index terms for that document
  - Assign a "weight" to each term based on its "importance"
  - Disregard order, structure, meaning, etc. of the words

What's a "word"? We'll return to this in a few lectures…

Sample Document

McDonald's slims down spuds

Fast-food chain to reduce certain types of fat in its french fries with new cooking oil.

NEW YORK (CNN/Money) - McDonald's Corp. is cutting the amount of "bad" fat in its french fries nearly in half, the fast-food chain said Tuesday as it moves to make all its fried menu items healthier.

But does that mean the popular shoestring fries won't taste the same? The company says no. "It's a win-win for our customers because they are getting the same great french-fry taste along with an even healthier nutrition profile," said Mike Roberts, president of McDonald's USA.

But others are not so sure. McDonald's will not specifically discuss the kind of oil it plans to use, but at least one nutrition expert says playing with the formula could mean a different taste.

Shares of Oak Brook, Ill.-based McDonald's (MCD: down $0.54 to $23.22, Research, Estimates) were lower Tuesday afternoon. It was unclear Tuesday whether competitors Burger King and Wendy's International (WEN: down $0.80 to $34.91, Research, Estimates) would follow suit. Neither company could immediately be reached for comment.

…

16 × said

14 × McDonalds

12 × fat

11 × fries

8 × new

6 × company french nutrition

5 × food oil percent reduce taste Tuesday

…

"Bag of Words"

What's the point?

- Retrieving relevant information is hard!
  - Evolving, ambiguous user needs, context, etc.
  - Complexities of language
- To operationalize information retrieval, we must vastly simplify the picture
- Bag-of-words approach:
  - Information retrieval is *all* (and *only*) about matching words in documents with words in queries
  - Obviously, not true…
  - But it works pretty well!

Why does "bag of words" work?

- Words alone tell us a lot about content

  Random: beating takes points falling another Dow 355

  Alphabetical: 355 another beating Dow falling points

  "Interesting": Dow points beating falling 355 another

  Actual: Dow takes another beating, falling 355 points

- It is relatively easy to come up with words that describe an information need

Vector Representation

- "Bags of words" can be represented as vectors
  - Why? Computational efficiency, ease of manipulation
  - Geometric metaphor: "arrows"
- A vector is a set of values recorded in any consistent order

"The quick brown fox jumped over the lazy dog's back"

→ [1 1 1 1 1 1 1 1 2]

1st position corresponds to "back"
2nd position corresponds to "brown"
3rd position corresponds to "dog"
4th position corresponds to "fox"
5th position corresponds to "jump"
6th position corresponds to "lazy"
7th position corresponds to "over"
8th position corresponds to "quick"
9th position corresponds to "the"

Representing Documents

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

Term	Document 1	Document 2
aid	0	1
all	0	1
back	1	0
brown	1	0
come	0	1
dog	1	0
fox	1	0
good	0	1
jump	1	0
lazy	1	0
men	0	1
now	0	1
over	1	0
party	0	1
quick	1	0
their	0	1
time	0	1

Stopword List

for
is
of
the
to

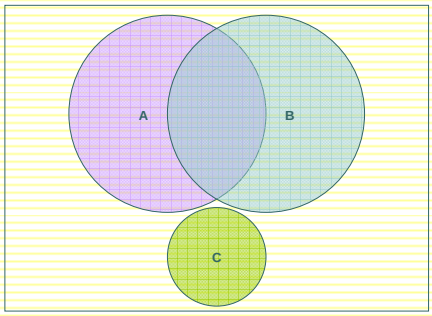
Boolean Retrieval

- Weights assigned to terms are either "0" or "1"
  - "0" represents "absence": term **isn't** in the document
  - "1" represents "presence": term **is** in the document
- Build queries by combining terms with Boolean operators
  - AND, OR, NOT
- The system returns all documents that satisfy the query

Why do we say that Boolean retrieval is "set-based"?

AND/OR/NOT

All documents



Logic Tables

A \ B	0	1
0	0	1
1	1	1

A OR B

B	0	1
	1	0

NOT B

A \ B	0	1
0	0	0
1	0	1

A AND B

A \ B	0	1
0	0	0
1	1	0

A NOT B
(= A AND NOT B)

Boolean View of a Collection

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8
aid	0	0	0	1	0	0	0	1
all	0	1	0	1	0	1	0	0
back	1	0	1	0	0	0	1	0
brown	1	0	1	0	1	0	1	0
come	0	1	0	1	0	1	0	1
dog	0	0	1	0	1	0	0	0
fox	0	0	1	0	1	0	1	0
good	0	1	0	1	0	1	0	1
jump	0	0	1	0	0	0	0	0
lazy	1	0	1	0	1	0	1	0
men	0	1	0	1	0	0	0	1
now	0	1	0	0	0	1	0	1
over	1	0	1	0	1	0	1	1
party	0	0	0	0	0	1	0	1
quick	1	0	1	0	0	0	0	0
their	1	0	0	0	1	0	1	0
time	0	1	0	1	0	1	0	0

Each column represents the view of a particular document: What terms are contained in this document?

Each row represents the view of a particular term: What documents contain this term?

To execute a query, pick out rows corresponding to query terms and then apply logic table of corresponding Boolean operator

Sample Queries

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8	
dog	0	0	1	0	1	0	0	0	
fox	0	0	1	0	1	0	1	0	
dog ∧ fox	0	0	1	0	1	0	0	0	dog AND fox → Doc 3, Doc 5
dog ∨ fox	0	0	1	0	1	0	1	0	dog OR fox → Doc 3, Doc 5, Doc 7
dog ¬ fox	0	0	0	0	0	0	0	0	dog NOT fox → empty
fox ¬ dog	0	0	0	0	0	0	1	0	fox NOT dog → Doc 7

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8	
good	0	1	0	1	0	1	0	1	
party	0	0	0	0	0	1	0	1	
g ∧ p	0	0	0	0	0	1	0	1	good AND party → Doc 6, Doc 8
over	1	0	1	0	1	0	1	1	
g ∧ p ¬ o	0	0	0	0	0	1	0	0	good AND party NOT over → Doc 6

Proximity Operators

- More "precise" versions of AND
  - "NEAR n" allows at most *n-1* intervening terms
  - "WITH" requires terms to be adjacent and in order
  - Other extensions: within *n* sentences, within *n* paragraphs, etc.
- Relatively easy to implement, but less efficient
  - Store position information for each word in the document vectors
  - Perform normal Boolean computations, but treat WITH and NEAR as extra constraints

Proximity Operator Example

Term	Doc 1	Doc 2
aid	0	1 (13)
all	0	1 (6)
back	1 (10)	0
brown	1 (3)	0
come	0	1 (9)
dog	1 (9)	0
fox	1 (4)	0
good	0	1 (7)
jump	1 (5)	0
lazy	1 (8)	0
men	0	1 (8)
now	0	1 (1)
over	1 (6)	0
party	0	1 (16)
quick	1 (2)	0
their	0	1 (15)
time	0	1 (4)

time AND come → Doc 2

time (NEAR 2) come → empty

quick (NEAR 2) fox → Doc 1

quick WITH fox → empty

Other Extensions

- Ability to search on fields
  - Leverage document structure: title, headings, etc.
- Wildcards
  - lov* = love, loving, loves, loved, etc.
- Special treatment of dates, names, companies, etc.

WESTLAW® Query Examples

- What is the statute of limitations in cases involving the federal tort claims act?
  - LIMIT! /3 STATUTE ACTION /S FEDERAL /2 TORT /3 CLAIM
- What factors are important in determining what constitutes a vessel for purposes of determining liability of a vessel owner for injuries to a seaman under the "Jones Act" (46 USC 688)?
  - (741 +3 824) FACTOR ELEMENT STATUS FACT /P VESSEL SHIP BOAT /P (46 +3 688) "JONES ACT" /P INJUR! /S SEAMAN CREWMAN WORKER
- Are there any cases which discuss negligent maintenance or failure to maintain aids to navigation such as lights, buoys, or channel markers?
  - NOT NEGLECT! FAIL! NEGLIG! /5 MAINT! REPAIR! /P NAVIGAT! /5 AID EQUIP! LIGHT BUOY "CHANNEL MARKER"
- What cases have discussed the concept of excusable delay in the application of statutes of limitations or the doctrine of laches involving actions in admiralty or under the "Jones Act" or the "Death on the High Seas Act"?
  - EXCUS! /3 DELAY /P (LIMIT! /3 STATUTE ACTION) LACHES /P "JONES ACT" "DEATH ON THE HIGH SEAS ACT" (46 +3 761)

Why Boolean Retrieval Works

- Boolean operators *approximate* natural language
  - Find documents about a good party that is not over
- AND can discover relationships between concepts
  - good party
- OR can discover alternate terminology
  - excellent party, wild party, etc.
- NOT can discover alternate meanings
  - Democratic party

The Perfect Query Paradox

- Every information need has a perfect set of documents
  - If not, there would be no sense doing retrieval
- Every document set has a perfect query
  - AND every word in a document to get a query for it
  - Repeat for each document in the set
  - OR every document query to get the set query
- But can users realistically be expected to formulate this perfect query?
  - Boolean query formulation is hard!

Why Boolean Retrieval Fails

- Natural language is way more complex
- AND "discovers" nonexistent relationships
  - Terms in different sentences, paragraphs, …
- Guessing terminology for OR is hard
  - good, nice, excellent, outstanding, awesome, …
- Guessing terms to exclude is even harder!
  - Democratic party, party to a lawsuit, …

Strengths and Weaknesses

- Strengths
  - Precise, if you know the right strategies
  - Precise, if you have an idea of what you're looking for
  - Efficient for the computer
- Weaknesses
  - Users must learn Boolean logic
  - Boolean logic insufficient to capture the richness of language
  - No control over size of result set: either too many documents or none
  - When do you stop reading? All documents in the result set are considered "equally good"
  - What about partial matches? Documents that "don't quite match" the query may be useful also

Ranked Retrieval

- Order documents by how likely they are to be relevant to the information need
  - Present hits one screen at a time
  - At any point, users can continue browsing through ranked list or reformulate query
- Attempts to retrieve relevant documents directly, not merely provide tools for doing so

Why Ranked Retrieval?

- Arranging documents by relevance is
  - Closer to how humans think: some documents are "better" than others
  - Closer to user behavior: users can decide when to stop reading
- Best (partial) match: documents need not have all query terms
  - Although documents with more query terms should be "better"
- Easier said than done!

A First Try

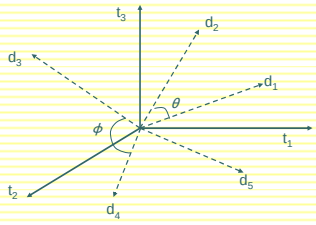
- Form several result sets from one long query
  - Query for the first set is the AND of all the terms
  - Then all but the first term, all but the second term, …
  - Then all but the first two terms, …
  - And so on until each single term query is tried
- Remove duplicates from subsequent sets
- Display the sets in the order they were made

Is there a more principled way to do this?

Similarity-Based Queries

- Let's replace relevance with "similarity"
  - Rank documents by their similarity with the query
- Treat the query as if it were a document
  - Create a query bag-of-words
- Find its similarity to each document
- Rank order the documents by similarity
- Surprisingly, this works pretty well!

Vector Space Model



Postulate: Documents that are "close together" in vector space "talk about" the same things

Therefore, retrieve documents based on how close the document is to the query (i.e., similarity ~ "closeness")

Similarity Metric

- How about $|d_1 - d_2|$?
  - This is the Euclidean distance between the vectors

Why is this not a good idea?

- Instead of distance, use "angle" between the vectors:

$$\cos(\theta) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|}$$

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

Components of Similarity

- The "inner product" (aka dot product) is the key to the similarity function

$$\vec{d}_j \cdot \vec{d}_k = \sum_{i=1}^{n} w_{i,j} w_{i,k}$$

Example: $[1\ 2\ 3\ 0\ 2] \cdot [2\ 0\ 1\ 0\ 2]$

$$= 1\times 2 + 2\times 0 + 3\times 1 + 0\times 0 + 2\times 2 = 9$$

- The denominator handles document length normalization

$$\left|\vec{d}_j\right| = \sqrt{\sum_{i=1}^{n} w_{i,k}^2}$$

Example: $\left|[1\ 2\ 3\ 0\ 2]\right|$

$$= \sqrt{1+4+9+0+4} = \sqrt{18} \approx 4.24$$

Reexamining Similarity

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

Query Vector

Inner Product

Length Normalization

Document Vector

How do we weight doc terms?

- Here's the intuition:
  - Terms that appear often in a document should get high weights
    - The more often a document contains the term "dog", the more likely that the document is "about" dogs.
  - Terms that appear in many documents should get low weights
    - Words like "the", "a", "of" appear in (nearly) all documents.
- How do we capture this mathematically?
  - Term frequency
  - Inverse document frequency

TF.IDF Term Weighting

- Simple, yet effective!

$$w_{i,j} = \mathrm{tf}_{i,j} \cdot \log \frac{N}{n_i}$$

$w_{i,j}$ **weight assigned to term *i* in document *j***

$\mathrm{tf}_{i,j}$ **number of occurrence of term *i* in document *j***

N **number of documents in entire collection**

n_i **number of documents with term *i***

TF.IDF Example

	tf				idf	$W_{i,j}$			
	1	2	3	4		1	2	3	4
complicated			5	2	0.301			1.51	0.60
contaminated	4	1	3		0.125	0.50	0.13	0.38	
fallout	5		4	3	0.125	0.63		0.50	0.38
information	6	3	3	2	0.000				
interesting		1			0.602		0.60		
nuclear	3		7		0.301	0.90		2.11	
retrieval		6	1	4	0.125		0.75	0.13	0.50
siberia	2				0.602	1.20			

Normalizing Document Vectors

- Recall our similarity function:

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

- Normalize document vectors in advance
  - Use the "cosine normalization" method: divide each term weight through by length of vector

Normalization Example

	tf				idf	$W_{i,j}$				$W'_{i,j}$			
	1	2	3	4		1	2	3	4	1	2	3	4
complicated			5	2	0.301			1.51	0.60			0.57	0.69
contaminated	4	1	3		0.125	0.50	0.13	0.38		0.29	0.13	0.14	
fallout	5		4	3	0.125	0.63		0.50	0.38	0.37		0.19	0.44
information	6	3	3	2	0.000								
interesting		1			0.602		0.60				0.62		
nuclear	3		7		0.301	0.90		2.11		0.53		0.79	
retrieval		6	1	4	0.125		0.75	0.13	0.50		0.77	0.05	0.57
siberia	2				0.602	1.20				0.71			
Length →						1.70	0.97	2.67	0.87				

Retrieval Example

Query: contaminated retrieval

$W'_{i,j}$

	query	1	2	3	4
complicated				0.57	0.69
contaminated	1	0.29	0.13	0.14	
fallout		0.37		0.19	0.44
information					
interesting			0.62		
nuclear		0.53		0.79	
retrieval	1		0.77	0.05	0.57
siberia		0.71			
similarity score →		0.29	0.9	0.19	0.57

Ranked list:
Doc 2
Doc 4
Doc 1
Doc 3

Do we need to normalize the query vector?

Weighted Retrieval

Weight query terms by assigning different term weights to query vector

Query: contaminated(3) retrieval

$w'_{i,j}$

	query	1	2	3	4
complicated				0.57	0.69
contaminated	3	0.29	0.13	0.14	
fallout		0.37		0.19	0.44
information					
interesting			0.62		
nuclear		0.53		0.79	
retrieval	1		0.77	0.05	0.57
siberia		0.71			
similarity score		0.87	1.16	0.47	0.57

Ranked list:
Doc 2
Doc 1
Doc 4
Doc 3

What's the point?

- Information seeking behavior is incredibly complex
- In order to build actual systems, we must make many simplifications
  - Absolutely unrealistic assumptions!
  - But the resulting systems are nevertheless useful
- Know what these limitations are!

Summary

- Boolean retrieval is powerful in the hands of a trained searcher
- Ranked retrieval is preferred in other circumstances
- Key ideas in the vector space model
  - Goal: find documents most similar to the query
  - Geometric interpretation: measure similarity in terms of angles between vectors in high dimensional space
  - Documents weights are some combinations of TF, DF, and Length
  - Length normalization is critical
  - Similarity is calculated via the inner product

One Minute Paper

- What was the muddiest point in today's class?