Ranked Retrieval

LBSC 796/INFM 718R

Session 3

February 16, 2011

Agenda

- Ranked retrieval
- Similarity-based ranking
- Probability-based ranking

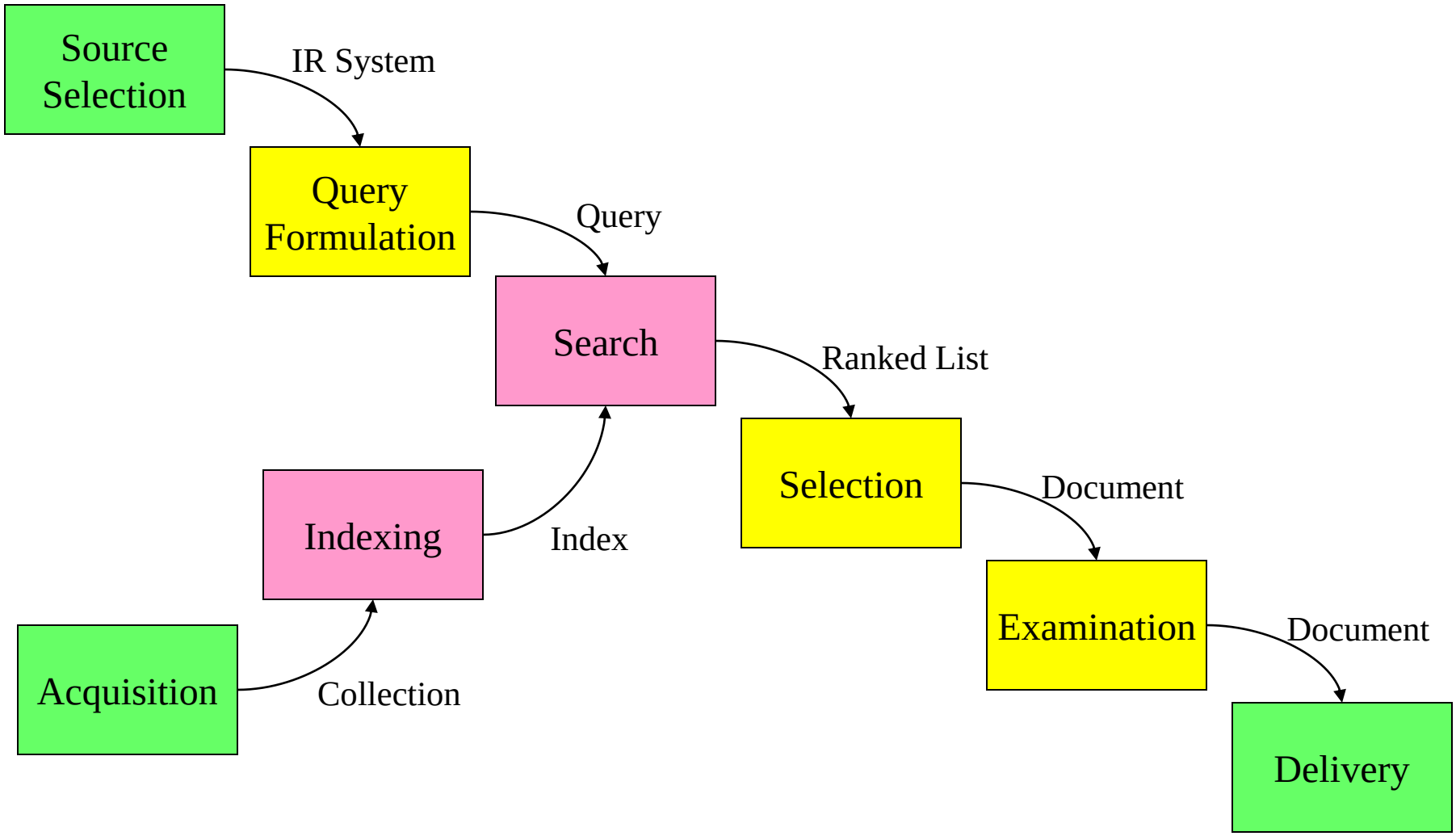
The Perfect Query Paradox

- Every information need has a perfect result set
  - All the relevant documents, no others
- Every result set has a (nearly) perfect query
  - AND every word to get a query for document 1
    - Use AND NOT for every other known word
  - Repeat for each document in the result set
  - OR them to get a query that retrieves the result set

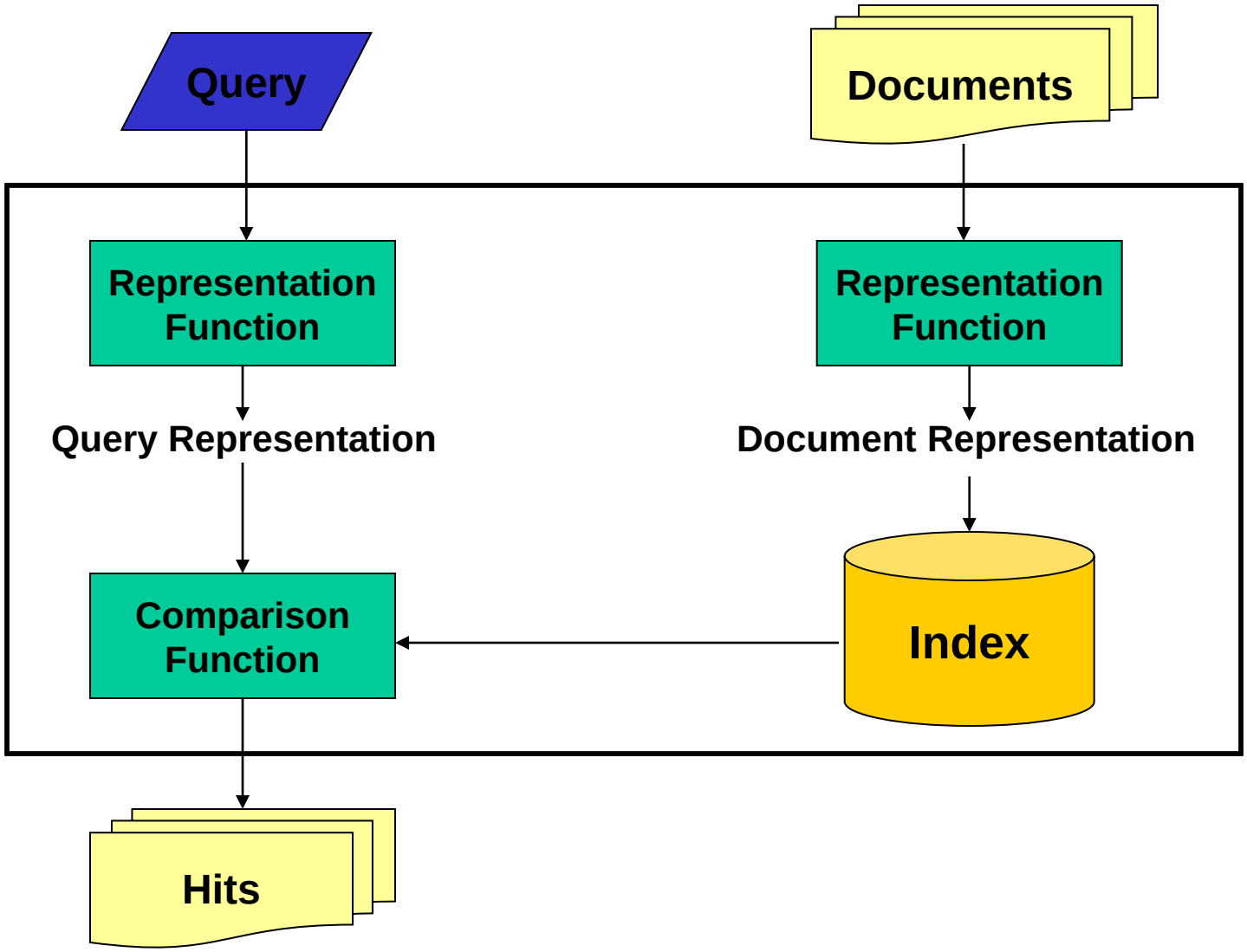
Boolean Retrieval

- Strong points
  - Accurate, **if you know the right strategies**
  - Efficient for the computer
- Weaknesses
  - **Often results in too many documents, or none**
  - **Users must learn Boolean logic**
  - Sometimes finds relationships that don't exist
  - Words can have many meanings
  - Choosing the right words is sometimes hard

Leveraging the User



Where Ranked Retrieval Fits



Ranked Retrieval Paradigm

- Perform a fairly general search
  - One designed to retrieve more than is needed
- Rank the documents in “best-first” order
  - Where best means “most likely to be relevant”
- Display as a list of easily skimmed “surrogates”
  - E.g., snippets of text that contain query terms

Advantages of Ranked Retrieval

- Leverages human strengths, covers weaknesses
  - Formulating precise queries can be difficult
  - People are good at recognizing what they want
- Moves decisions from query to selection time
  - Decide how far down the list to go as you read it
- Best-first ranking is an understandable idea

Ranked Retrieval Challenges

- "Best first" is easy to say but hard to do!
  - Computationally, we can only approximate it
- Some details will be opaque to the user
  - Query reformulation requires more guesswork
- More expensive than Boolean
  - Storing evidence for "best" requires more space
  - Query processing time increases with query length

Simple Example: Partial-Match Ranking

- Form all possible result sets in this order:
  - AND all the terms to get the first set
  - AND all but the 1st term, all but the 2nd, …
  - AND all but the first two terms, …
  - And so on until every combination has been done
- Remove duplicates from subsequent sets
- Display the sets in the order they were made
  - Document rank within a set is arbitrary

Partial-Match Ranking Example

information AND retrieval

Readings in Information Retrieval
Information Storage and Retrieval
Speech-Based Information Retrieval for Digital Libraries
Word Sense Disambiguation and Information Retrieval

information NOT retrieval

The State of the Art in Information Filtering

retrieval NOT information

Inference Networks for Document Retrieval
Content-Based Image Retrieval Systems
Video Parsing, Retrieval and Browsing
An Approach to Conceptual Text Retrieval Using the EuroWordNet …
Cross-Language Retrieval: English/Russian/French

Agenda

- Ranked retrieval
- Similarity-based ranking
- Probability-based ranking

What's a Model?

- A construct to help understand a complex system
  - A particular way of "looking at things"

- Models inevitably make simplifying assumptions

Similarity-Based Queries

- Model relevance as “similarity”
  - Rank documents by their similarity to 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!

Similarity-Based Queries

- Treat the query as if it were a document
  - Create a query bag-of-words
- Find the similarity of each document
  - Using the coordination measure, for example
- Rank order the documents by similarity
  - Most similar to the query first
- Surprisingly, this works pretty well!
  - Especially for very short queries

Document Similarity

- How similar are two documents?
  - In particular, how similar is their bag of words?

1: Nuclear fallout contaminated Montana.

2: Information retrieval is interesting.

3: Information retrieval is complicated.

	1	2	3
complicated			1
contaminated	1		
fallout	1		
information		1	1
interesting		1	
nuclear	1		
retrieval		1	1
siberia	1		

The Coordination Measure

- Count the number of terms in common
  - Based on Boolean bag-of-words
- Documents 2 and 3 share two common terms
  - But documents 1 and 2 share no terms at all
- Useful for "more like this" queries
  - "more like doc 2" would rank doc 3 ahead of doc 1
- Where have you seen this before?

Coordination Measure Example

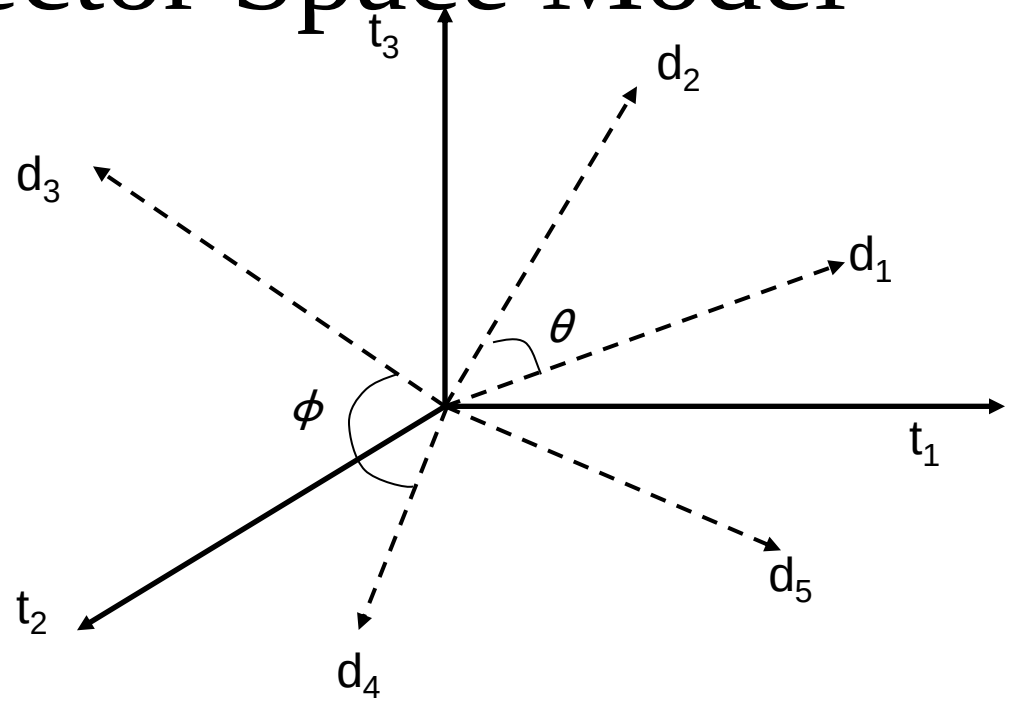
	1	2	3
complicated			1
contaminated	1		
fallout	1		
information		1	1
interesting		1	
nuclear	1		
retrieval		1	1
siberia	1		

Query: complicated retrieval
Result: 3, 2

Query: interesting nuclear fallout
Result: 1, 2

Query: information retrieval
Result: 2, 3

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")

Counting Terms

- Terms tell us about documents
  - If “rabbit” appears a lot, it may be about rabbits
- Documents tell us about terms
  - “the” is in every document -- not discriminating
- Documents are most likely described well by rare terms that occur in them frequently
  - Higher “term frequency” is stronger evidence
  - Low “document frequency” makes it stronger still

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

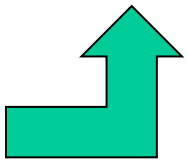
11 × fries

8 × new

6 × company, french, nutrition

5 × food, oil, percent, reduce, taste, Tuesday

…



“Bag of Words”

A Partial Solution: TF*IDF

- High TF is evidence of meaning
- Low DF is evidence of term importance
  - Equivalently high "IDF"
- Multiply them to get a "term weight"
- Add up the weights for each query term

Let N be the total number of documents

Let DF of the N documents contain term i

Let $TF_{i,j}$ be the number of times term i appears in document j

Then $w_{i,j} = TF_{i,j} \cdot \log \frac{N}{DF}$

TF*IDF Example

	$tf_{i,j}$ 1	2	3	4	idf_i	$w_{i,j}$ 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			

query:
contaminated retrieval
Result: 2, 3, 1, 4

The Document Length Effect

- People want documents with useful parts
  - But scores are computed for the whole document
- Document lengths vary in many collections
  - So frequency could yield inconsistent resutls
- Two strategies
  - Adjust term frequencies for document length
  - Divide the documents into equal “passages”

Document Length Normalization

- Long documents have an unfair advantage
  - They use a lot of terms
    - So they get more matches than short documents
  - And they use the same words repeatedly
    - So they have much higher term frequencies
- Normalization seeks to remove these effects
  - Related somehow to maximum term frequency

“Cosine” Normalization

- Compute the length of each document vector
  - Multiply each weight by itself
  - Add all the resulting values
  - Take the square root of that sum
- Divide each weight by that length

Let $w_{i,j}$ be the unnormalized weight of term *i* in document *j*

Let $w'_{i,j}$ be the normalized weight of term *i* in document *j*

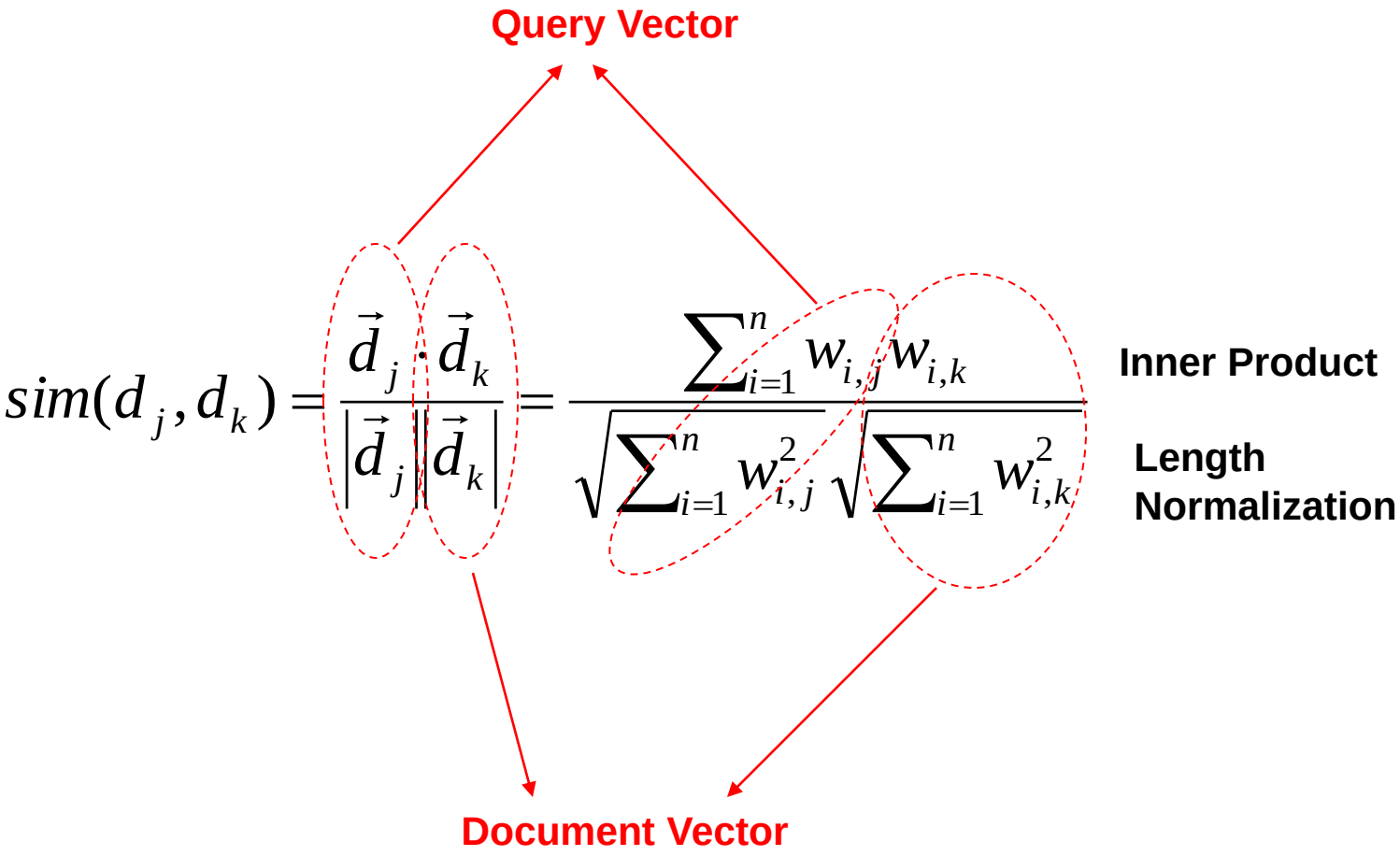
Then $w'_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_j w_{i,j}^2}}$

Cosine Normalization Example

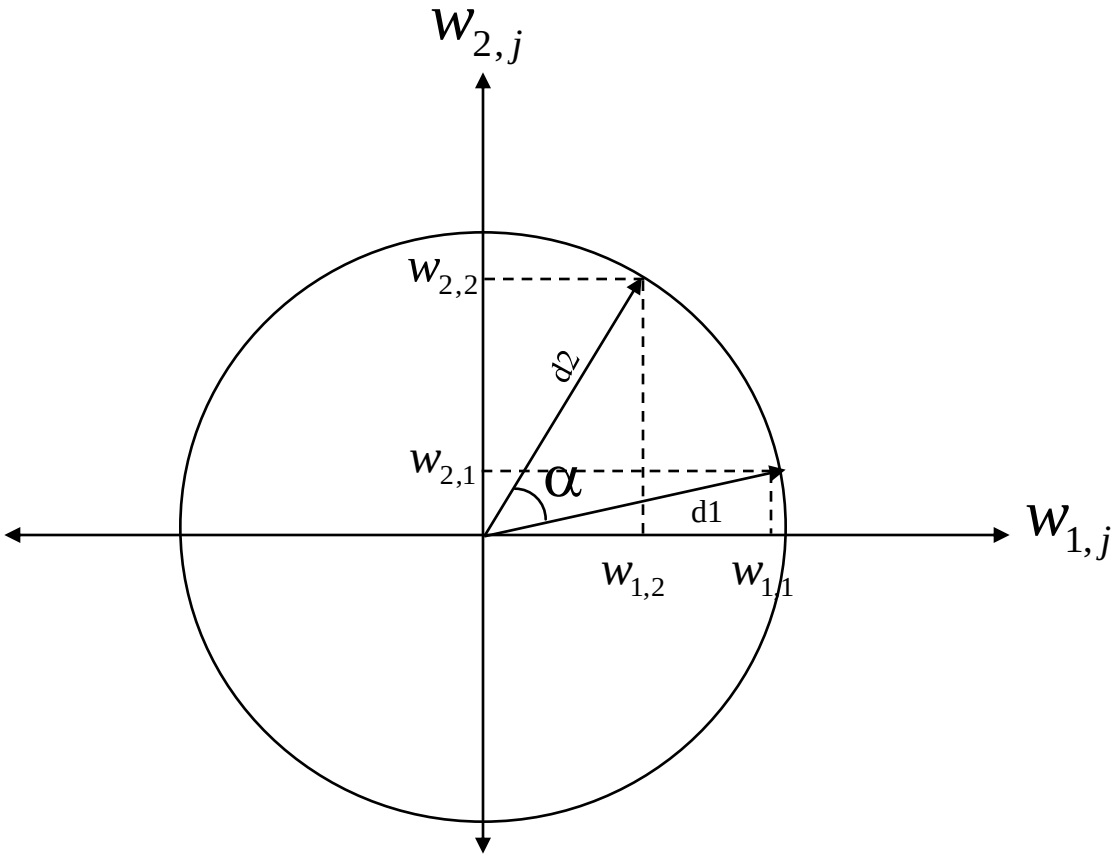
	$tf_{i,j}$				idf_i	$w_{i,j}$				$w'_{i,j}$			
	1	2	3	4		1	2	3	4	1	2	3	4
complicated			5	2	0.301			1.51	0.60			0.57	0.69
contaminated	4	1	3		0.125	0.50	0.13	0.38		0.29	0.13	0.14	
fallout	5		4	3	0.125	0.63		0.50	0.38	0.37		0.19	0.44
information	6	3	3	2	0.000								
interesting		1			0.602		0.60				0.62		
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				

query: contaminated retrieval, Result: 2, 4, 1, 3 (compare to 2, 3, 1, 4)

Formally …


Query Vector
$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{|\vec{d}_j||\vec{d}_k|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$
Inner Product
Length Normalization
Document Vector

Why Call It "Cosine"?



Let document 1 have unit length with coordinates $w_{1,1}$ and $w_{2,1}$

Let document 2 have unit length with coordinates $w_{1,2}$ and $w_{2,2}$

Then $\cos\alpha = \sqrt{(w_{1,1} \cdot w_{1,2}) + (w_{2,1} \cdot w_{2,2})}$

Interpreting the Cosine Measure

- Think of query and the document as vectors
  - Query normalization does not change the ranking
  - Square root does not change the ranking
- Similarity is the angle between two vectors
  - Small angle = very similar
  - Large angle = little similarity
- Passes some key sanity checks
  - Depends on pattern of word use but not on length
  - Every document is most similar to itself

"Okapi BM-25" Term Weights

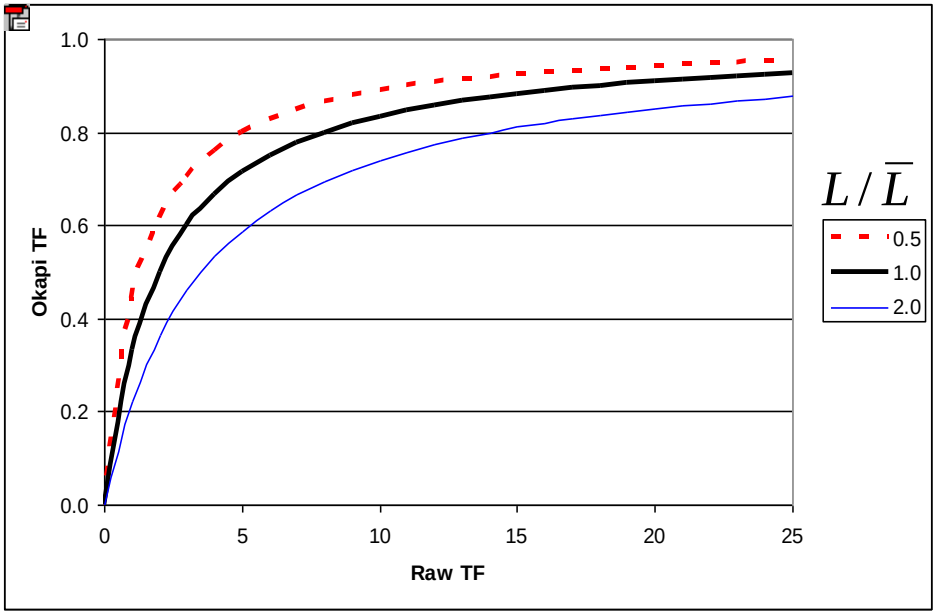
Let L_i be the number of terms in document i

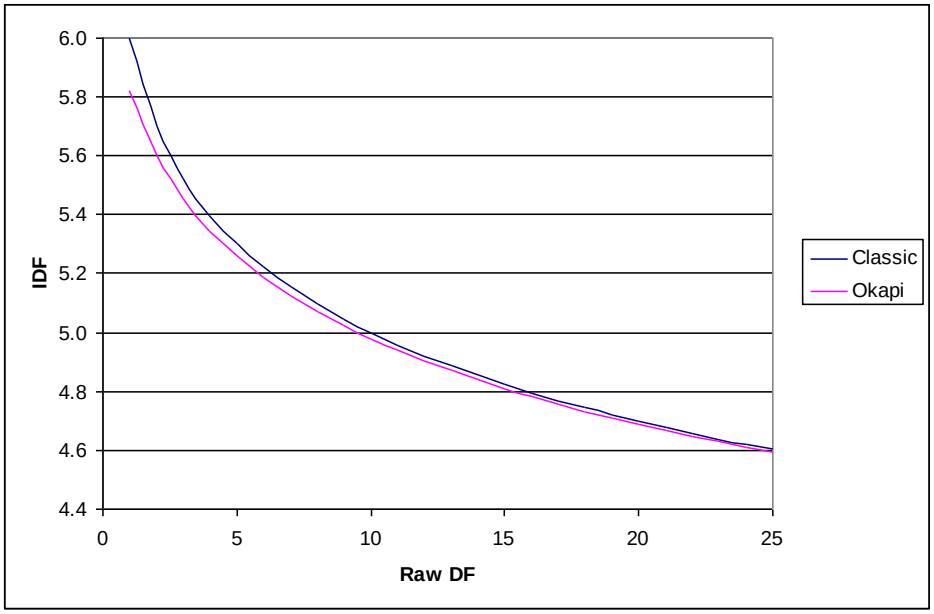
Let $\overline{L}$ be the average number of terms in a document

$$w_{i,j} = \frac{TF_{i,j}}{1.5\frac{L_i}{\overline{L}} + TF_{i,j} + 0.5} * \log\left(\frac{N - DF_j + 0.5}{DF_j + 0.5}\right)$$

TF component

IDF component





BM25F

- Identify fields
  - Title, body, anchor text
- Assign weights
  - Learned on training data
- Combine term frequencies
  - Rescale k1

Learning To Rank

- Feature engineering
  - Content, popularity, accessibility, centrality, …
- Parameterized function
  - λ, k1, b, …
- Evaluation measure
  - P@10, MAP, NDCG, …
- Supervised learning
  - Relevance judgments
  - Training, Devtest, Evaluation

Passage Retrieval

- Another approach to long-document problem
  - E.g., break it up into coherent units
- Recognizing topic boundaries can be hard
  - Overlapping 300 word passages work well
- Use best passage rank as the document's rank
  - Passage ranking can also help focus examination

Summary

- Goal: find documents most similar to the query
- Compute normalized document term weights
  - Some combination of TF, DF, and Length
- Sum the weights for each query term
  - In linear algebra, this is an “inner product” operation

Agenda

- Ranked retrieval
- Similarity-based ranking
- ➢ Probability-based ranking

The Key Idea

- We ask "is this document relevant?"
  - Vector space: we answer "somewhat"
  - Probabilistic: we answer "probably"

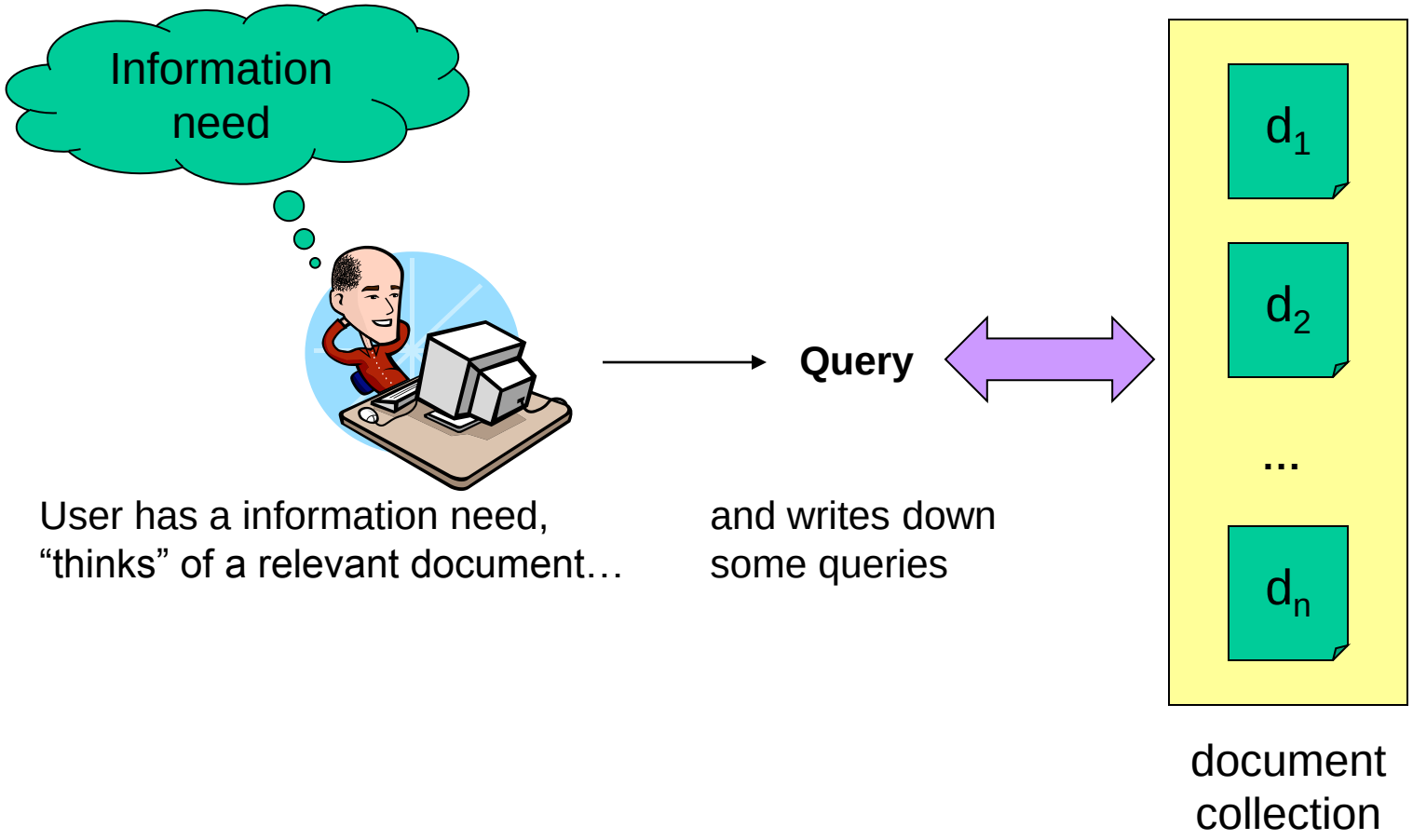
- The key is to know what "probably" means
  - First, we'll formalize that notion
  - Then we'll apply it to ranking

Probability Ranking Principle

- Assume binary relevance, document independence
  - Each document is either relevant or it is not
  - Relevance of one doc reveals nothing about another
- Assume the searcher works down a ranked list
  - Seeking some number of relevant documents
- Then it follows that:
  - Documents should be ranked in order of decreasing probability of relevance to the query,

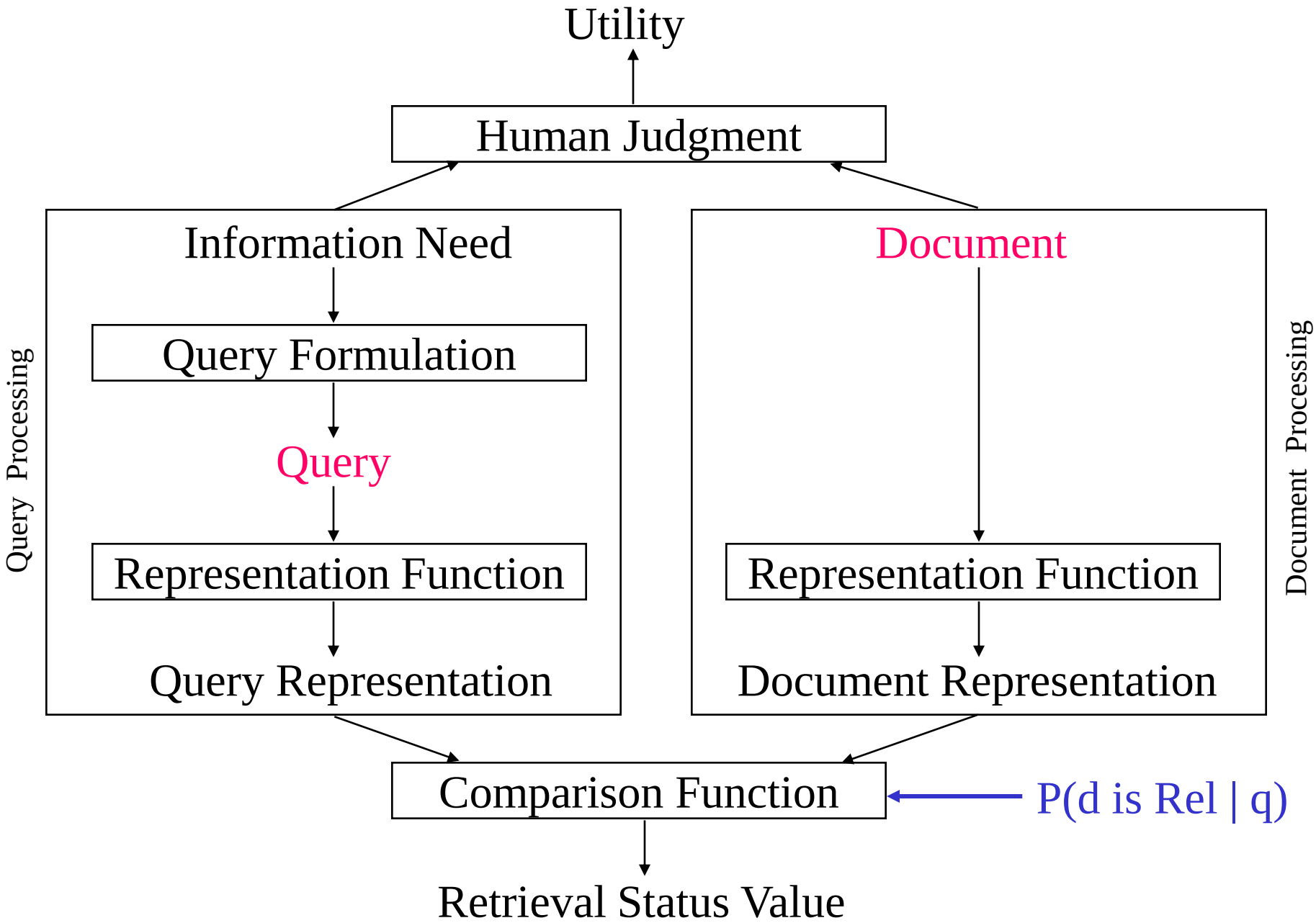
    P(d relevant-to q)

“Noisy-Channel” Model of IR



Information retrieval: given the query, guess the document it came from.

Where do the probabilities fit?



Probability

- Alternative definitions
  - Statistical: relative frequency as $n \to \infty$
  - Subjective: degree of belief

- Thinking statistically
  - Imagine a finite amount of "stuff"
  - Associate the number 1 with the total amount
  - Distribute that "mass" over the possible events

Statistical Independence

- A and B are independent if and only if:

  $P(A \text{ and } B) = P(A) \times P(B)$

- Independence formalizes "unrelated"
  - P("being brown eyed") = 85/100
  - P("being a doctor") = 1/1000
  - P("being a brown eyed doctor") = 85/100,000

Dependent Events

- Suppose”
  - P(“having a B.S. degree”) = 2/10
  - P(“being a doctor”) = 1/1000
- Would you expect
  - P(“having a B.S. degree and being a doctor”)
    = 2/10,000 ???
- Extreme example:
  - P(“being a doctor”) = 1/1000
  - P(“having studied anatomy”) = 12/1000

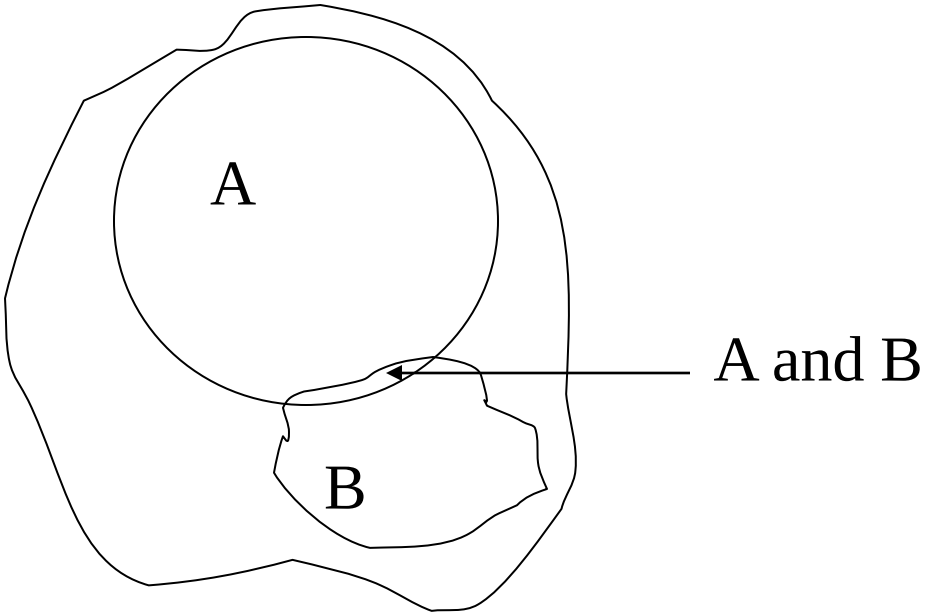
More on Conditional Probability

- Suppose
  - P("having studied anatomy") = 12/1000
  - P("being a doctor and having studied anatomy") = 1/1000
- Consider
  - P("being a doctor" | "having studied anatomy") = 1/12
- But if you assume all doctors have studied anatomy
  - P("having studied anatomy" | "being a doctor") = 1

Useful restatement of definition: $P(A \text{ and } B) = P(A|B) \times P(B)$

Conditional Probability

- P(A | B) ≡ P(A and B) / P(B)



- P(A) = prob of A relative to the whole space
- P(A|B) = prob of A considering only the cases where B is known to be true

Some Notation

- Consider
  - A set of hypotheses: H1, H2, H3
  - Some observable evidence O

- P(O|H1) = probability of O being observed if we *knew* H1 were true
- P(O|H2) = probability of O being observed if we *knew* H2 were true
- P(O|H3) = probability of O being observed if we *knew* H3 were true

An Example

- Let
  - O = “Joe earns more than $100,000/year”
  - H1 = “Joe is a doctor”
  - H2 = “Joe is a college professor”
  - H3 = “Joe works in food services”
- Suppose we do a survey and we find out
  - P(O|H1) = 0.6
  - P(O|H2) = 0.07
  - P(O|H3) = 0.001
- What should be our guess about Joe’s profession?

Bayes' Rule

- What's P(H1|O)? P(H2|O)? P(H3|O)?
- Theorem:

$$P(H \mid O) = \frac{P(O \mid H) \times P(H)}{P(O)}$$

Posterior probability

Prior probability

- Notice that the prior is very important!

Back to the Example

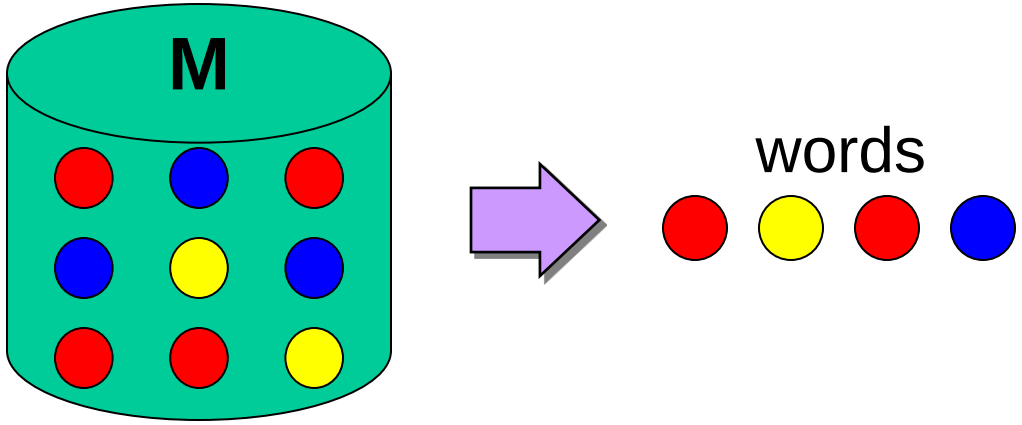
- Suppose we also have good data about priors:
  - P(O|H1) = 0.6 P(H1) = 0.0001 doctor
  - P(O|H2) = 0.07 P(H2) = 0.001 prof
  - P(O|H3) = 0.001 P(H3) = 0.2 food
- We can calculate
  - P(H1|O) = 0.00006 / P("earning > $100K/year")
  - P(H2|O) = 0.0007 / P("earning > $100K/year")
  - P(H3|O) = 0.0002 / P("earning > $100K/year")

Key Ideas

- Defining probability using frequency
- Statistical independence
- Conditional probability
- Bayes' rule

A “Language Model”

- Colored balls are randomly drawn from an urn (with replacement)



P(● ● ● ●) = P(●)× P(●) ×P(●) ×P(●)

= (4/9) × (2/9) × (4/9) × (3/9)

Comparing Language Models

Model M_1

P(w)	w
0.2	the
0.0001	yon
0.01	class
0.0005	maiden
0.0003	sayst
0.0001	pleaseth
…	

Model M_2

P(w)	w
0.2	the
0.1	yon
0.001	class
0.01	maiden
0.03	sayst
0.02	pleaseth
…	

the	class	pleaseth	yon	maiden
0.2	0.01	0.0001	0.0001	0.0005
0.2	0.001	0.02	0.1	0.01

$$P(s|M_2) > P(s|M_1)$$

What exactly does this mean?

Retrieval w/ Language Models

- Build a model for every document
- Rank document *d* based on $P(M_D \mid q)$
- Expand using Bayes' Theorem

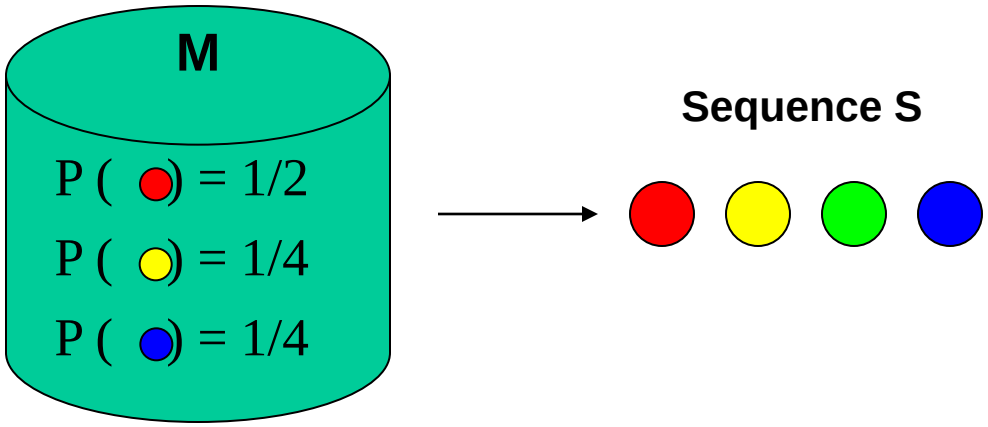
$$P(M_D \mid q) = \frac{P(q \mid M_D)P(M_D)}{P(q)}$$

P(q) is same for all documents; doesn't change ranks
P(M_D) [the prior] is assumed to be the same for all *d*

- Same as ranking by $P(q \mid M_D)$

Zero-Frequency Problem

- Suppose some event is not in our observation S
  - Model will assign zero probability to that event

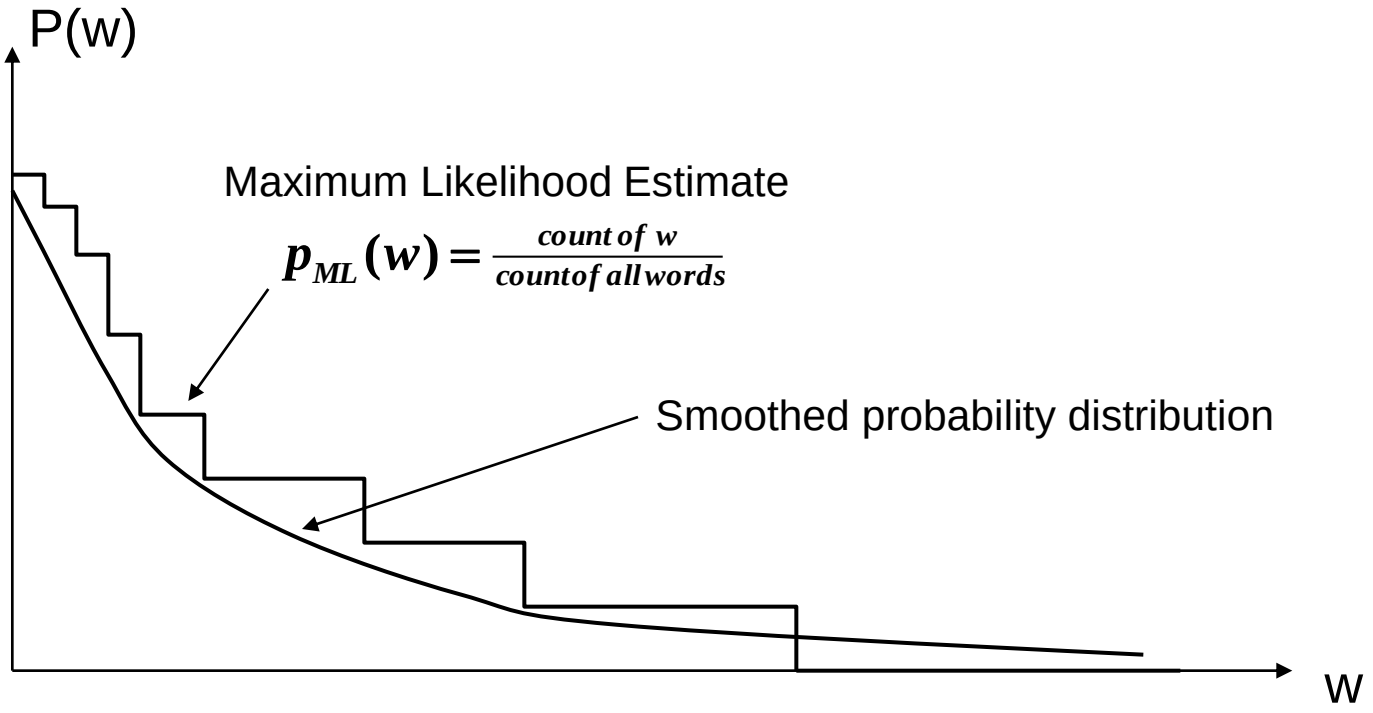


$$P(\bullet\ \bullet\ \bullet\ \bullet) = P(\bullet) \times P(\bullet) \times P(\bullet) \times P(\bullet)$$

$$= (1/2) \times (1/4) \times 0 \times (1/4) = 0$$!!

Smoothing

The solution: "smooth" the word probabilities



Recap: LM for IR

- Indexing-time:
  - Build a language model for every document
- Query-time Ranking
  - Estimate the probability of generating the query according to each model
  - Rank the documents according to these probabilities

Language Model Advantages

- Conceptually simple
- Explanatory value
- Exposes assumptions
- Minimizes reliance on heuristics

Key Ideas

- Probabilistic methods formalize assumptions
  - Binary relevance
  - Document independence
  - Term independence
  - Uniform priors
  - Top-down scan
- Natural framework for combining evidence
  - e.g., non-uniform priors

A Critique

- Most of the assumptions are not satisfied!
  - Searchers want utility, not relevance
  - Relevance is not binary
  - Terms are clearly not independent
  - Documents are often not independent
- Smoothing techniques are somewhat *ad hoc*

But It Works!

- Ranked retrieval paradigm is powerful
  - Well suited to human search strategies
- Probability theory has explanatory power
  - At least we know where the weak spots are
  - Probabilities are good for combining evidence
- Good implementations exist (e.g., Lemur)
  - Effective, efficient, and large-scale

Comparison With Vector Space

- Similar in some ways
  - Term weights based on frequency
  - Terms often used as if they were independent
- Different in others
  - Based on probability rather than similarity
  - Intuitions are probabilistic rather than geometric

One Minute Paper

- Which assumption underlying the probabilistic retrieval model causes you the most concern, and why?