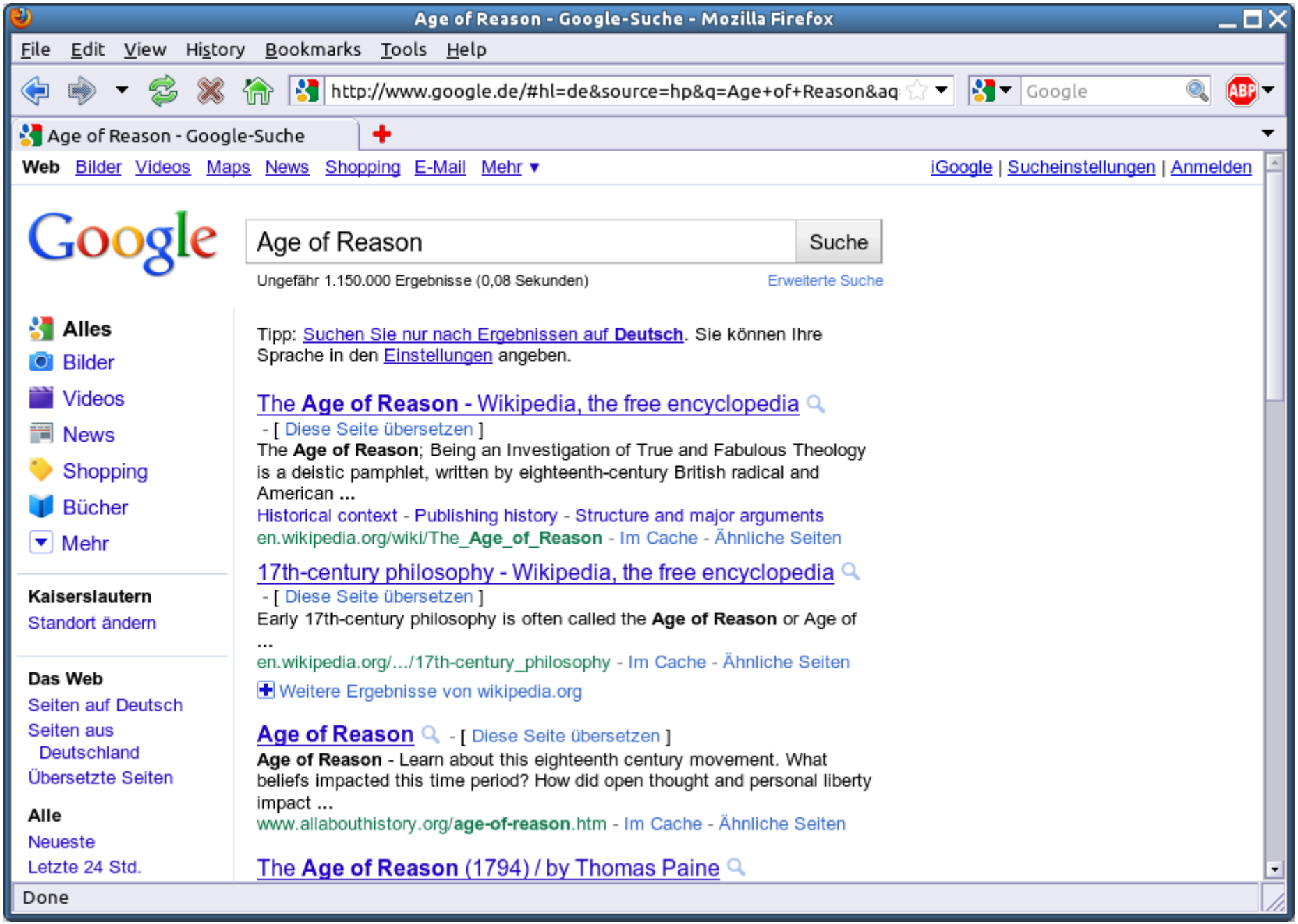
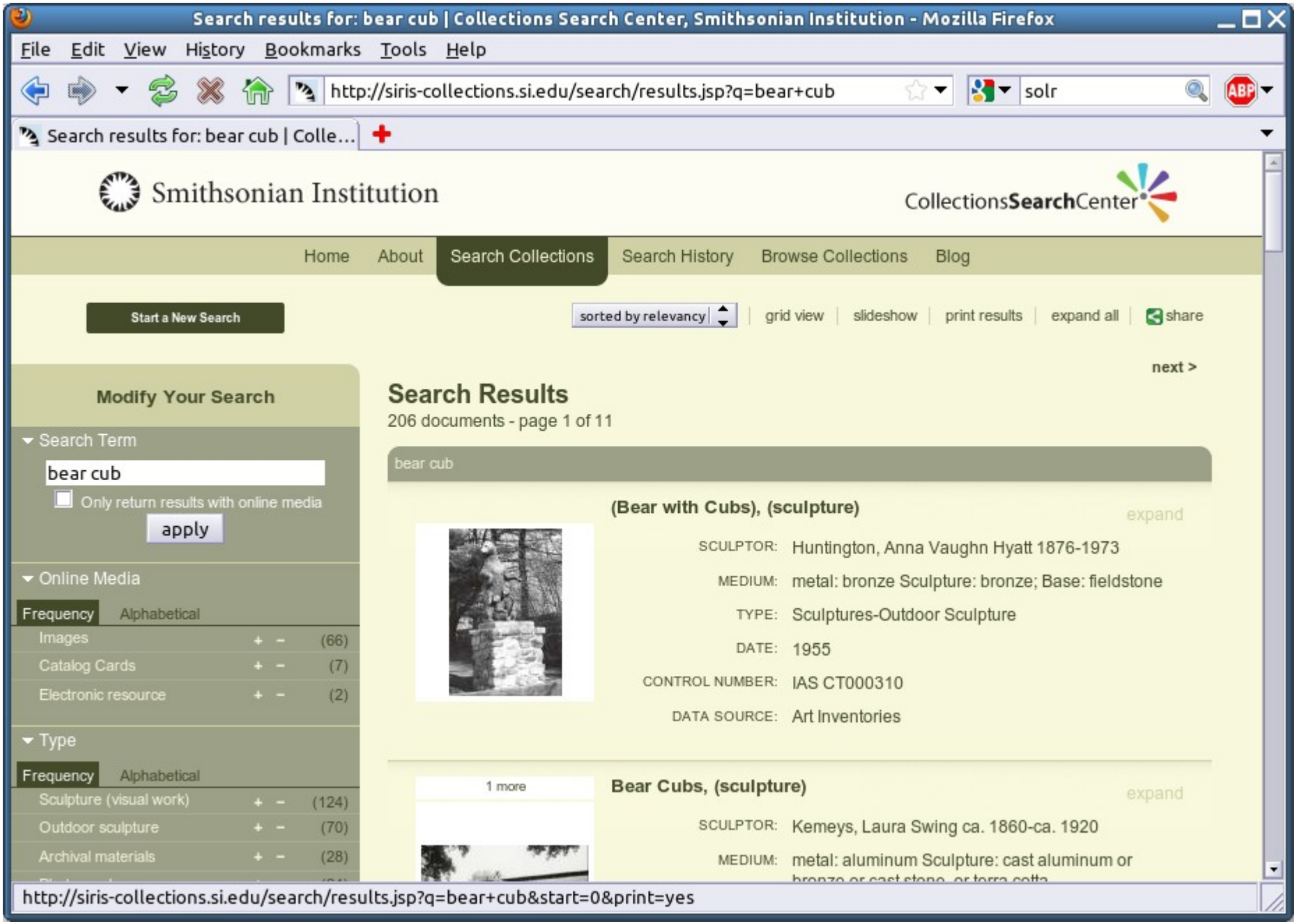
Text Information Retrieval

Prof. Thomas Breuel


Age of Reason - Google-Suche - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.google.de/#hl=de&source=hp&q=Age+of+Reason&aq
Google
Age of Reason - Google-Suche
Web Bilder Videos Maps News Shopping E-Mail Mehr ▾
iGoogle | Sucheinstellungen | Anmelden
Google
Age of Reason
Suche
Ungefähr 1.150.000 Ergebnisse (0,08 Sekunden)
Erweiterte Suche
Alles
Bilder
Videos
News
Shopping
Bücher
Mehr
Kaiserslautern
Standort ändern
Das Web
Seiten auf Deutsch
Seiten aus Deutschland
Übersetzte Seiten
Alle
Neueste
Letzte 24 Std.
Tipp: Suchen Sie nur nach Ergebnissen auf Deutsch. Sie können Ihre Sprache in den Einstellungen angeben.
The Age of Reason - Wikipedia, the free encyclopedia
- [Diese Seite übersetzen]
The Age of Reason; Being an Investigation of True and Fabulous Theology is a deistic pamphlet, written by eighteenth-century British radical and American ...
Historical context - Publishing history - Structure and major arguments
en.wikipedia.org/wiki/The_Age_of_Reason - Im Cache - Ähnliche Seiten
17th-century philosophy - Wikipedia, the free encyclopedia
- [Diese Seite übersetzen]
Early 17th-century philosophy is often called the Age of Reason or Age of ...
en.wikipedia.org/.../17th-century_philosophy - Im Cache - Ähnliche Seiten
Weitere Ergebnisse von wikipedia.org
Age of Reason - [Diese Seite übersetzen]
Age of Reason - Learn about this eighteenth century movement. What beliefs impacted this time period? How did open thought and personal liberty impact ...
www.allabouthistory.org/age-of-reason.htm - Im Cache - Ähnliche Seiten
The Age of Reason (1794) / by Thomas Paine
Done



http://en.wikipedia.org/wiki/Solr http://siris-collections.si.edu/search/results.jsp?q=bear+cub

EVALUATION

information retrieval

- **input:** query Q
- **output:** (depending on system)
  - single result D_i
  - set of results $S_i = \{ D_{i1} , \ldots , D_{ik} \}$
  - list of ranked results $S_i = [D_{i1} , \ldots , D_{ik}]$
- **ground truth:**
  - relevant result D^*_i
  - set of relevant results S^*_i

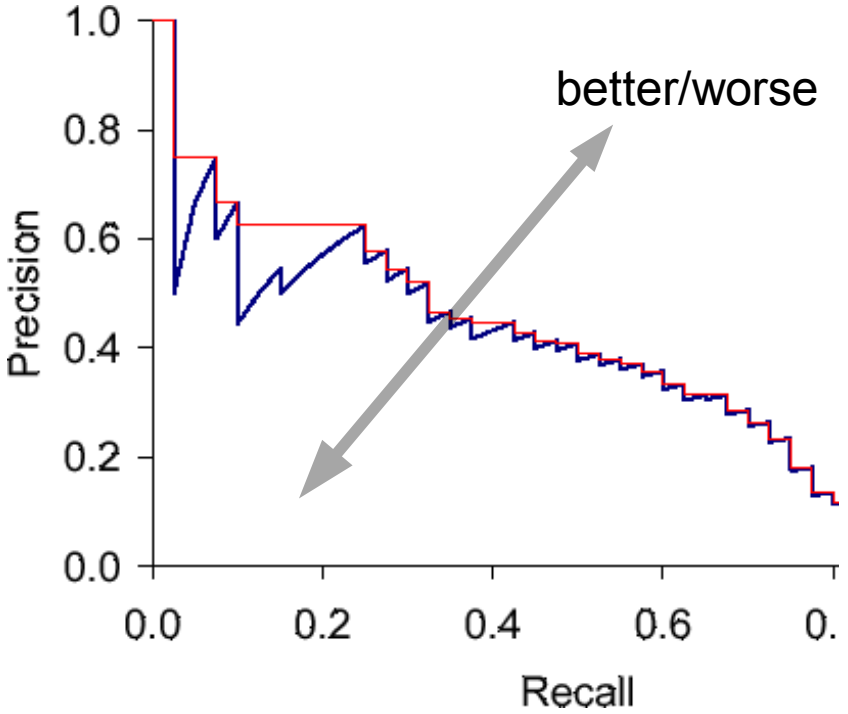
evaluating relevance

- **for now, assume set-based results**
  - given a query, return a set of results (no ranking)
- **for each query**
  - manually determine relevance (yes/no) of each document
  - compute precision and recall (below)

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

precision/recall graph based on ranks



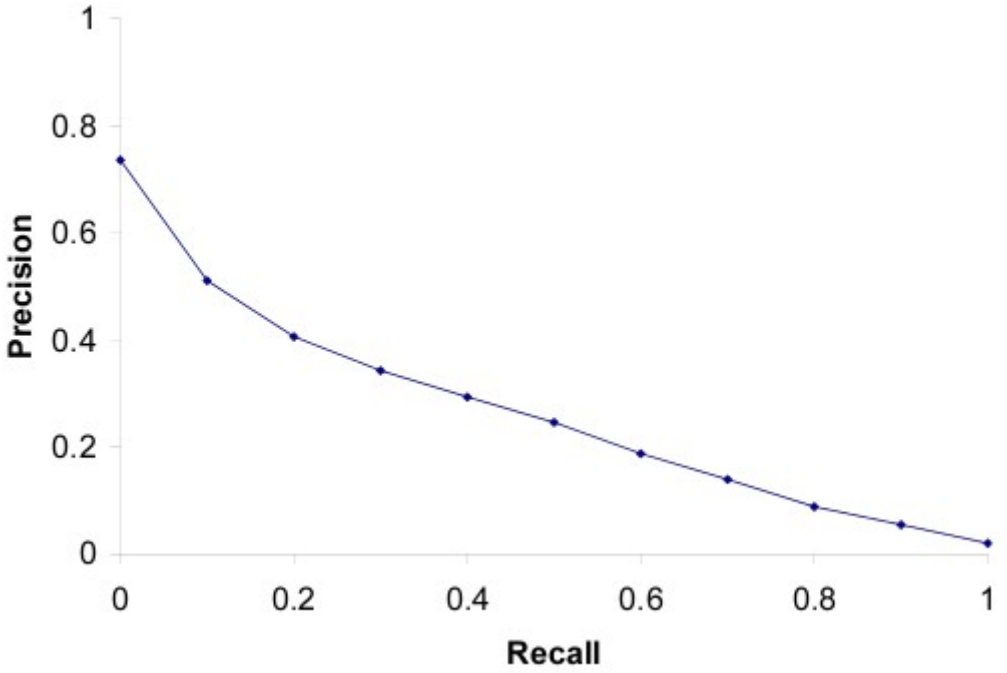
1	2	3	4	5	6	7	8	9
1	0	0	1	1	0	0	0	0

cutoff = 4
precision = 0.5
recall = 0.66

- compute P/R cutting of at different ranks (precision-at-n)
- sawtooth from correct/incorrect results
- usually interpolate these curves (red)
- need to choose a tradeoff between precision and recall
- multifactorial optimization
- (other kinds of parameters also possible)

http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

11-point p/r graph, 50 queries



- evaluated at 11 points (0.0, 0.1, … 1.0)
- averaged over 50 queries

F-measure

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}.$$

$$F_\beta = \frac{(1 + \beta^2) \cdot (\text{precision} \cdot \text{recall})}{(\beta^2 \cdot \text{precision} + \text{recall})}$$

- **F_β measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision**
- **Weighted harmonic mean of precision and recall.**
- **Harmonic mean: progress first at rate 1 then at rate 2 same as progress at harmonic mean of the two rates.**

Mean Average Precision

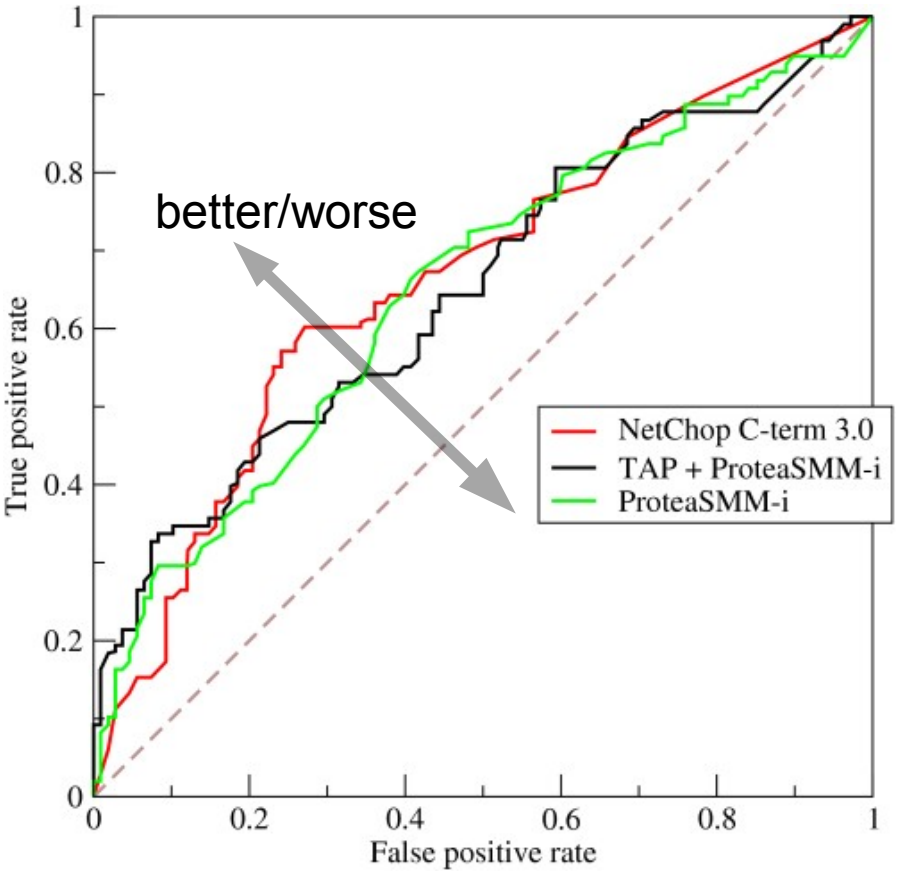
- **people want a single number for competitions**
- **average precision**
  - area under the precision/recall curve
  - often computed at just 11 points
- **mean average precision (MAP)**
  - given a set of queries...
  - compute the average precision score for each query
  - compute the mean of those average precisions

error classes

		correct result / classification	
		E1	**E2**
obtained result / classification	**E1**	**tp** (true positive)	**fp** (false positive)
	E2	**fn** (false negative)	**tn** (true negative)

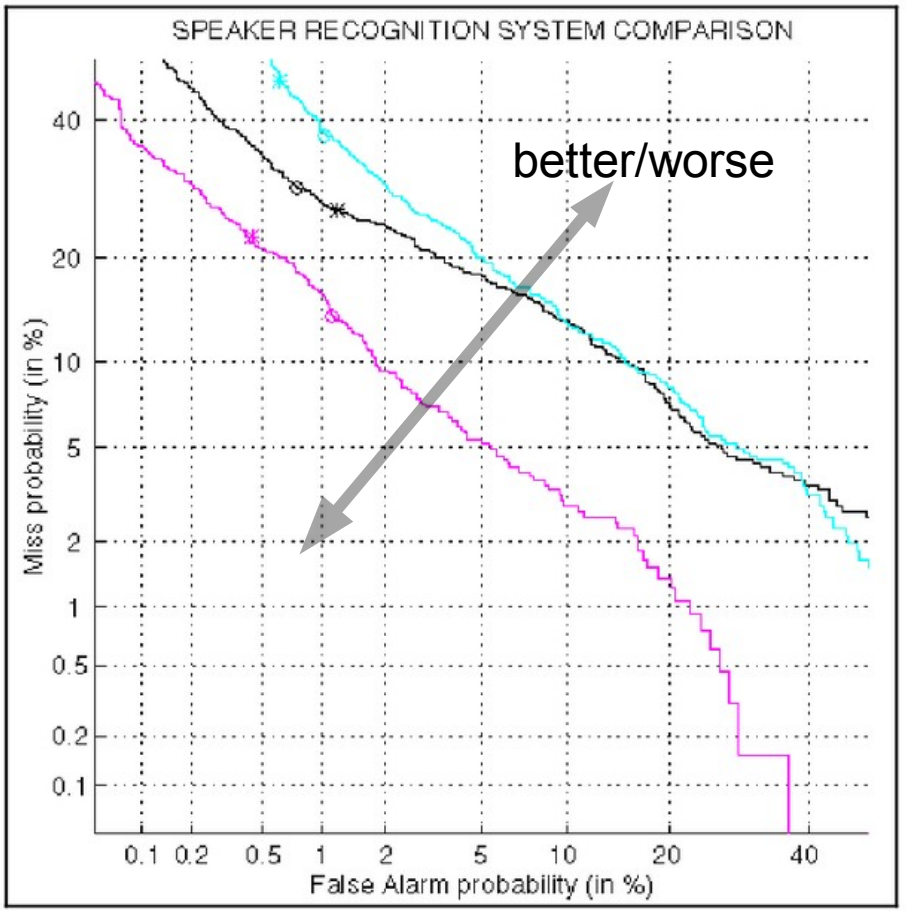
$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

ROC curve



- similar to precision/recall curve, but TP vs FP
- different axes
- multifactorial optimization as well
- relationship explored in paper
- curves can *dominate* each other
- don't work well in IR because FP usually climbs very rapidly

DET curve



- detection error tradeoff curve
- note logarithmic scale
- FN vs FP plot

evaluation

- **be able to explain**
  - test set
  - rank vs set-based evaluation
  - precision, recall, TP, FP, …
  - F-score
  - MAP
  - tradeoffs, curves, dominance
  - precision/recall curve, ROC curve, DET curve

VECTOR SPACE MODEL

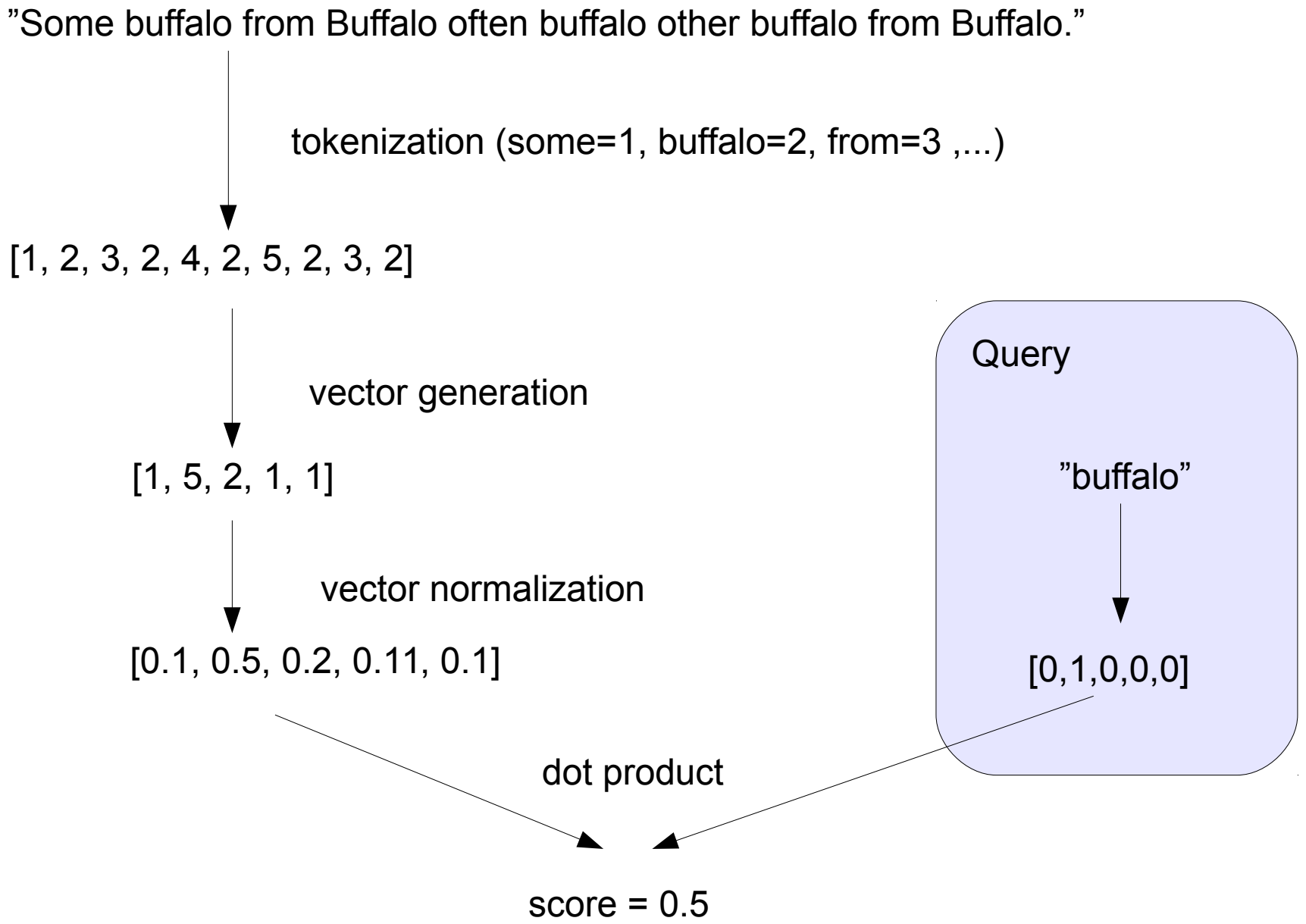
text retrieval

- **data**
  - queries Q are lists/sets/multisets of words
  - documents D_i are lists/sets/multisets of words
- **how do we find relevant documents?**
  - find all documents for which $Q \cap D_i \neq \emptyset$
  - find all documents for which $Q \cap D_i = Q$
  - find all documents for which $|Q \cap D_i|$ is maximal
- **(can you write SQL statements to do this?)**

vector space model

- **representation**
  - transform sets Q and D_i into vectors q and v_i
  - return documents for which $d(q,v_i)>t$ or $d(q,v_i)$=maximal
  - usually $d(x,y) = x \cdot y / (|x| \, |y|)$ (*cosine similarity*)
- **transformation**
  - assign a number j to each word w_j (the *term vocabulary*)
  - $v_{ij} = 1$ if $w_j \in D_j$, else 0
  - v_{ij} = count of w_j in D_j
  - v_{ij} = f(count of w_j in D_j, size of D_j,...)

vector space model



TF-IDF model

- **specific method of assigning weights**
  - tf_{ij} : frequency of term i in document j
  - idf_i : log of inverse frequency of documents containing term i
  - $w_{ij} = tfidf_{ij} = tf_{ij} \cdot idf_i$
- **terms get high weights if...**
  - they occur frequently in the document
  - occur rarely in the database (logarithmically weighted)
- **works very well in practice**

some more details

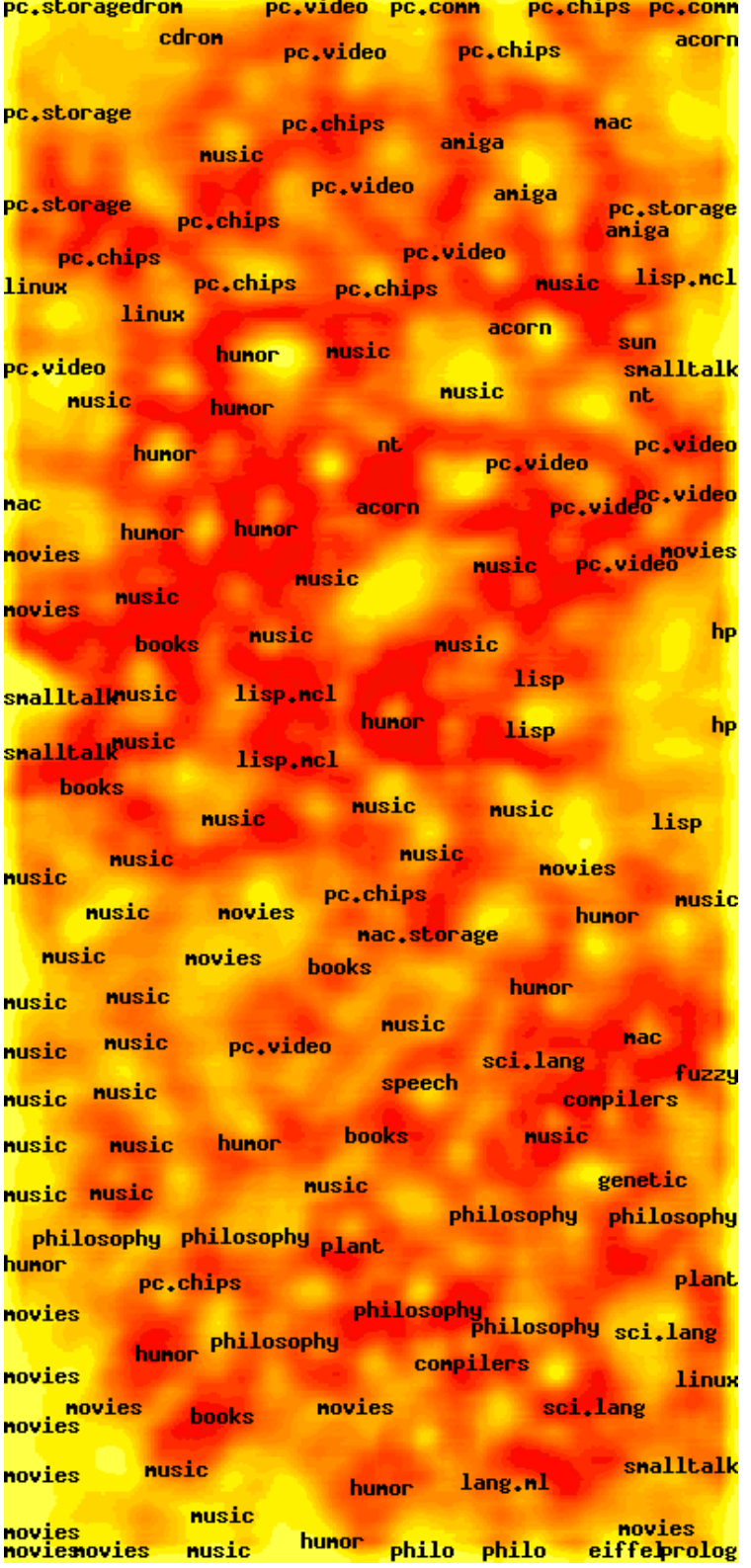
- **"document" might be...**
  - paragraph, page, etc.
  - granularity of indexing
- **tokenization...**
  - usually need morphological analysis
  - sow, sown, sowed, sows, sewn → "sow"
  - stemming, lemmatization, ...
  - finding word boundaries in languages w/o spaces
  - leave out frequent words (stop words, stop list)
  - equivalences ("little", "small")
  - case folding, accents, international characters, Unicode

comments on efficiency

- document vectors are very sparse (why?)
- naive representation wastes lots of space
- can use sparse vectors
- can compute all dot products in different order
- can transform into different spaces (e.g., via PCA)
- dot product $x \cdot y$ closely related to distance
- vector space retrieval closely related to nearest neighbor
- vector representation allows applying many patrec algs

document clustering

- **documents are (sparse) vectors**
- **perform clustering via...**
  - k-means algorithm
  - self-organizing map algorithm
  - hierarchical clustering
- **will talk about these algorithms later**



latent semantic analysis

- attempt to write all documents d_i as a linear combination of a small number of basis vectors b_j

$$d_i = \sum \lambda_{ij} b_j$$

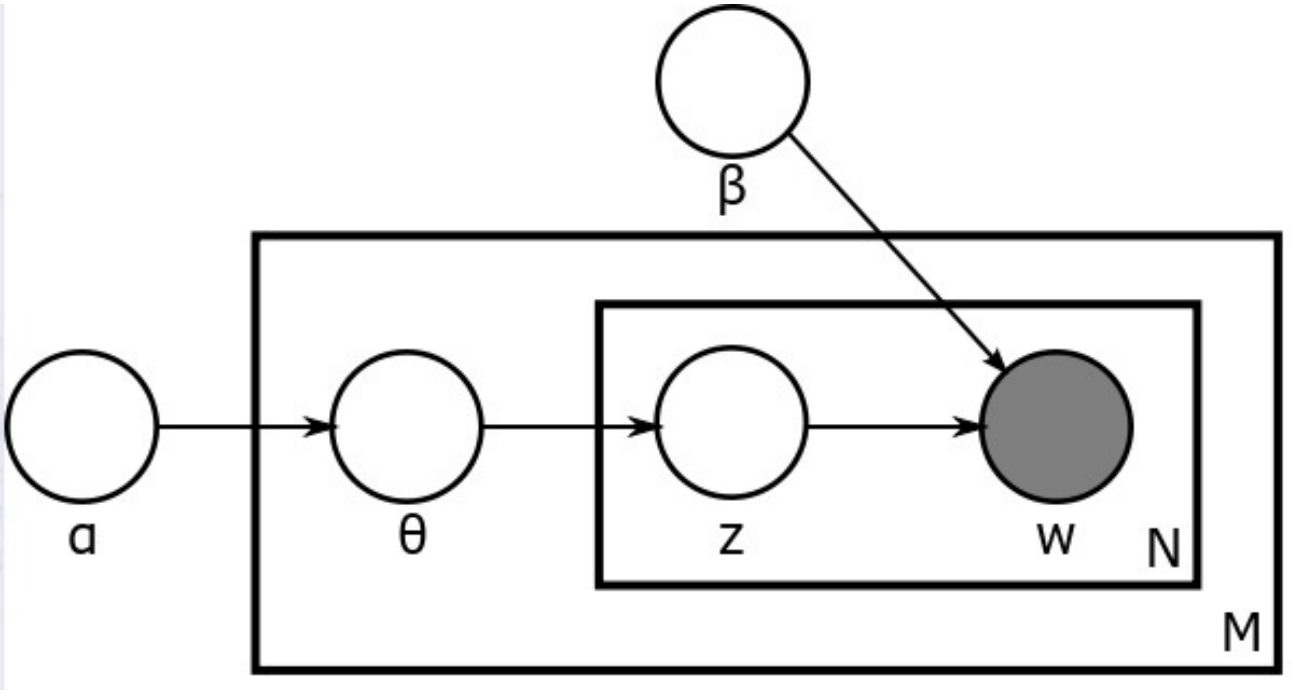
- idea: each basis vector represents a kind of "topic" or "subject", and each document is a combination of topics
- retrieve documents by similarity of the λ_i vectors
- pattern recognition: PCA, information retrieval: LSA

PROBABILISTIC MODELS

probabilistic model

- **generation of a document**
  - first pick one or more *topics* and how much they contribute
  - for each topic, randomly pick words according to that topic
- **parameters**
  - P(topic)
  - P(word|topic)
- **variables**
  - sample of topics for a given document (latent)
  - sample of words for a given topic (latent)
  - words in a given document (observable)

plate model



α is the parameter of the uniform Dirichlet prior on the per-document topic distributions.
β is the parameter of the uniform Dirichlet prior on the per-topic word distribution.
θ_i is the topic distribution for document i,
z_{ij} is the topic for the jth word in document i, and
w_{ij} is the specific word.

M: number of documents
N: number of words in document