IR1: Introduction to Information Retrieval

Some slides due to Raghavan et al.

Slides by Andrew DeOrio

Agenda

- What is search?
- Boolean retrieval
- Vector space model
- tf-idf
- Assessing rank quality
  - Precision/Recall Curves
  - Kendall’s Tau
  - Mean Reciprocal Rank

client

what sound do cows make?

server

Search backend

Magic: searches Internet

Moo!

ahead of time

1. Download Internet
2. Build an index

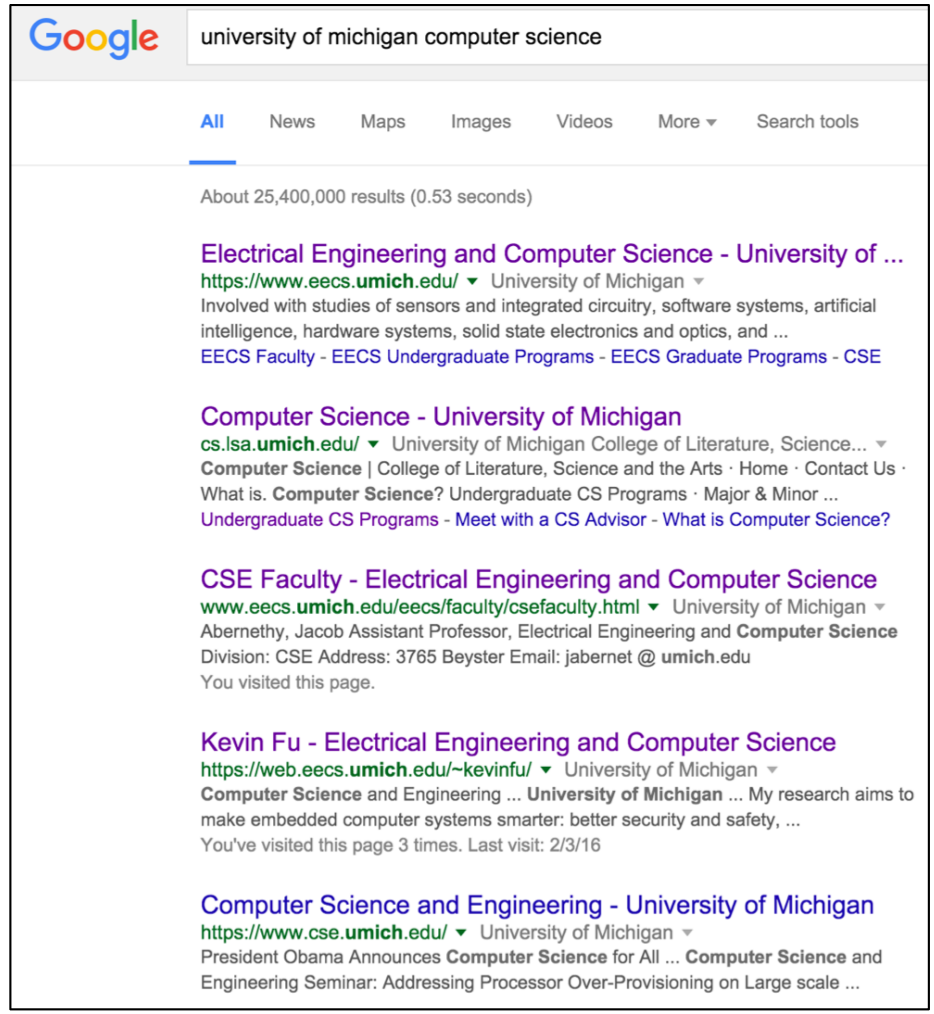
run time

1. Use index
2. Rank results
3. Do it fast.

Key problem: ranking results

- 33% clicks on top result

- Different ranking methods
  - Use words on page
    - Pre-Google
    - Today's lecture
  - Use links on page
    - Next time
    - Google



Agenda

- What is search?
- **Boolean retrieval**
- Vector space model
- tf-idf
- Assessing rank quality
  - Precision/Recall Curves
  - Kendall's Tau
  - Mean Reciprocal Rank

Boolean retrieval

- For each doc, two possible outcomes of query processing
  - TRUE or FALSE
  - "exact match" retrieval
  - Simplest form of search, used to be common

Query

- Which plays of Shakespeare contain the words **Brutus** AND **Caesar** but NOT **Calpurnia**?
- Answer queries like this using a *term-document incidence table*
- Basically, a table of Booleans

Term-document incidence

1 if **play** contains **word**

Which plays of Shakespeare contain the words **Brutus** AND **Caesar** but NOT **Calpurnia**?

	Tempest	Hamlet	Othello	Macbeth
Antony	0	0	0	1
Brutus	0	1	0	0
Caesar	0	1	1	1
Calpurnia	0	0	0	0
Cleopatra	0	0	0	0
mercy	1	1	1	1
worser	1	1	1	0

Ranking search results

- Boolean queries simply *include* or *exclude* a document from results
- When we have many hits, we need to sort (rank) the results

Agenda

- What is search?
- Boolean retrieval
- **Vector space model**
- tf-idf
- Assessing rank quality
  - Precision/Recall Curves
  - Kendall's Tau
  - Mean Reciprocal Rank

Documents as vectors

- Each document is a vector of values, one component for each term
- We thus have a *vector space*
  - A doc is a point in the space
- How many dimensions?
- Is this space sparse or dense? Why?

Documents as vectors

- Each document is a vector of values, one component for each term

- We thus have a *vector space*
  - A doc is a point in the space

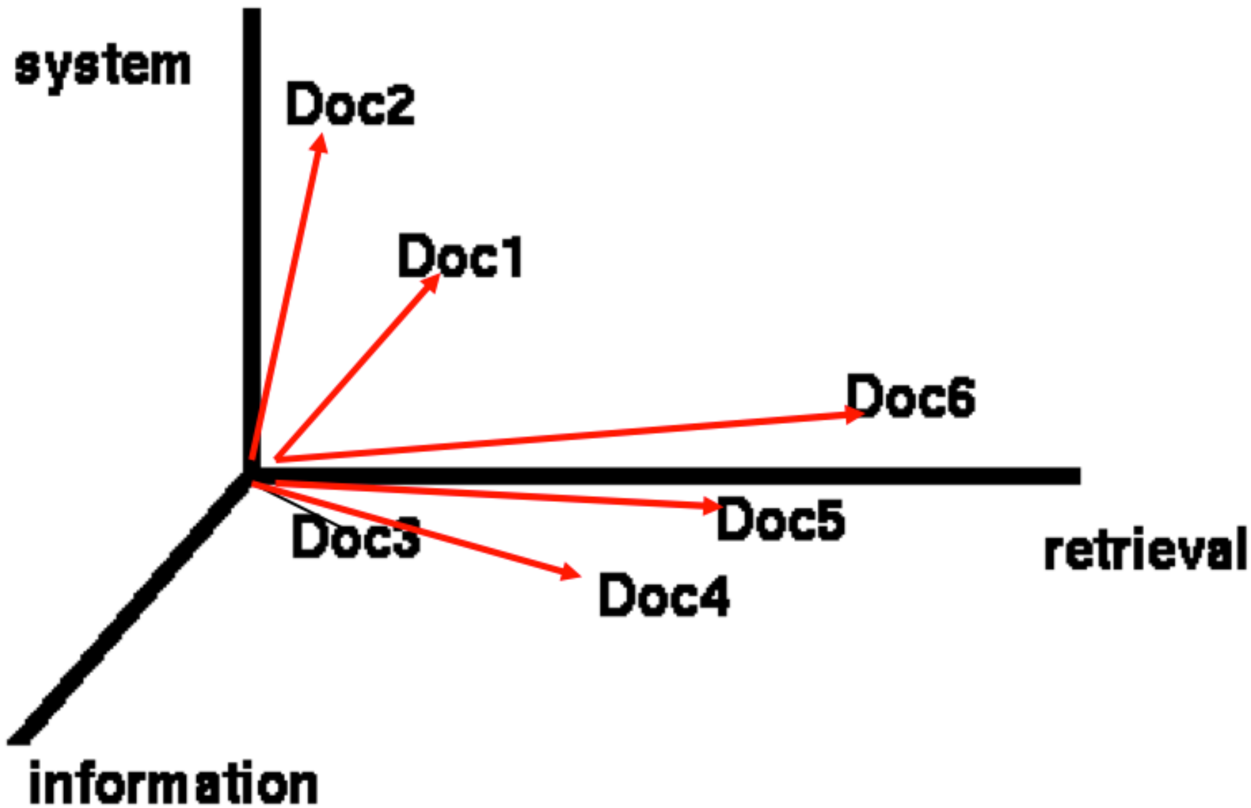
- How many dimensions?
  - Dimension for every possible term (word)
- Is this space sparse or dense? Why?
  - Sparse. Most documents do not have most words.

Documents in 3D space

- Documents that are "close together" in space are close in meaning

Documents in 3D space

- Documents that are "close together" in space are close in meaning



Vector space query model

1. Treat a query as a short document
2. Sort documents by increasing distance (decreasing similarity) to the query document
3. Easy to compute, both query and doc are vectors

- First used in Salton’s SMART system (1970). Now used by almost every information retrieval system.

Vector representation

- Docs and queries are vectors
- Position 1 corresponds to term 1
  Position *t* corresponds to term *t*
- Weight of term stored in each position

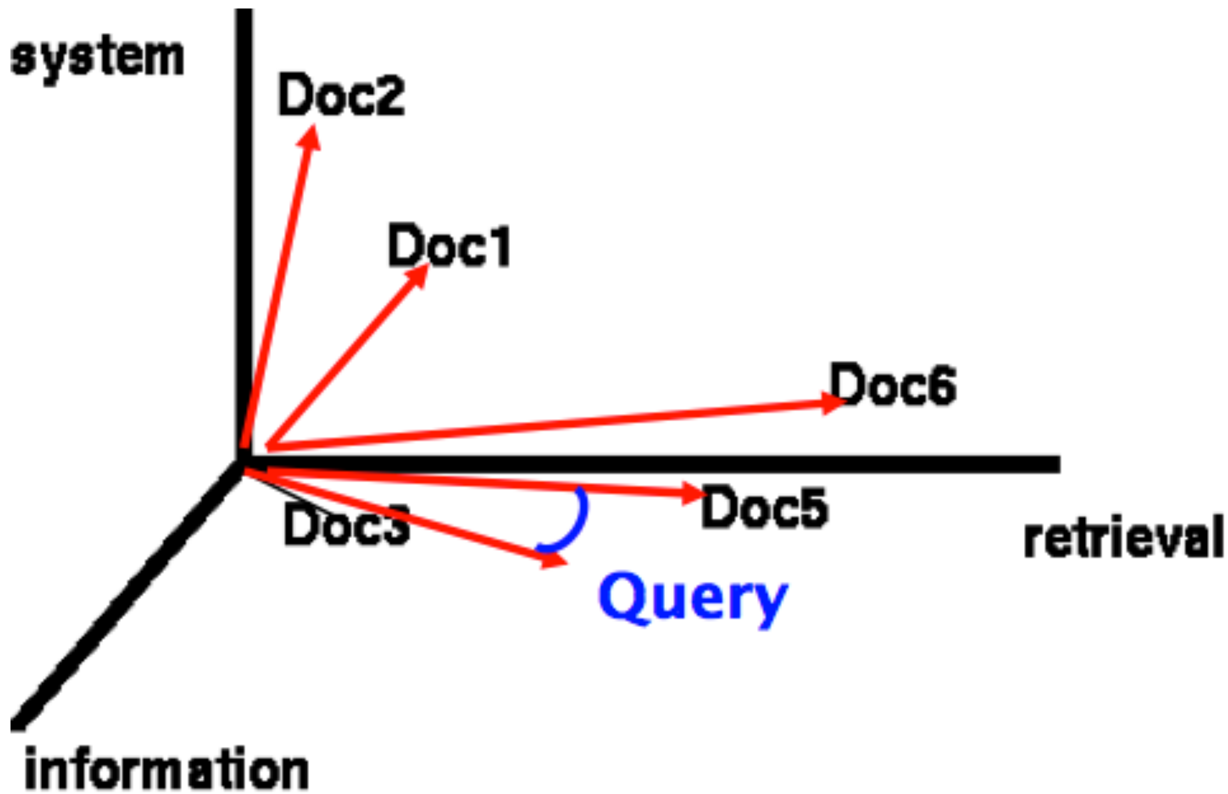
$$D_i = w_{d_{i1}}, w_{d_{i2}}, \ldots, w_{d_{it}}$$

$$Q = w_{q1}, w_{q2}, \ldots, w_{qt}$$

$$w = 0 \text{ if a term is absent}$$

Documents in 3D space

- Term weights indicate length of document vector along a dimension



Vector magnitude

- With Boolean search, the values in a vector (one for each term) are 0 or 1
- Next, we'll consider values [0, 1]

Agenda

- What is search?
- Boolean retrieval
- Vector space model
- **tf-idf**
- Assessing rank quality
  - Precision/Recall Curves
  - Kendall’s Tau
  - Mean Reciprocal Rank

Term frequency

- Which doc is a better match for the query "kangaroo", one with a single mention of "kangaroo", or a doc that mentions it 10 times?

- Find a few good examples and bad examples on the web for "kangaroo"
  - How many times does the word appear?

Term frequency

- Which doc is a better match for the query "kangaroo", one with a single mention of "kangaroo", or a doc that mentions it 10 times?
- The doc that mentions it 10 times
- *Term frequency:* how many times the word appears in current document
- Higher is better

Document frequency

- Which term is more indicative of document similarity, "Book" or "Rumpelstiltskin"?

Document frequency

- Which term is more indicative of document similarity, "Book" or "Rumpelstiltskin"?
- Rumpelstiltskin
- *Document frequency:* how often a word appears in doc collection
- Lower is better

Inverse document frequency

- Inverse Document Frequency (IDF) provides high values for rare words, low values for common words
- $IDF = N / n_k$
  - N = total # docs in collection *C*
  - n_k = # docs in *C* that contain T_k

Example

- Document collection: 29 million wikipedia pages
- "rumpelstiltskin" appears in 3600 wikipedia pages
- IDF = 29e6 / 3600 = 8055.556

Log inverse document frequency

- Query "Rumpelstiltskin book"
- What if "Rumpelstiltskin" appears 10 out of 29 million documents and "book" appears in 1 million out of 29 million?
- Is the term "book" 100,000 times less useful than "Rumpelstiltskin"?

Log inverse document frequency

- Query "Rumpelstiltskin book"
- What if "Rumpelstiltskin" appears 10 out of 29 million documents and "book" appears in 1 million out of 29 million?
- Is the term "book" 100,000 times less useful than "Rumpelstiltskin"?
  - Probably not.

- Solution: $\mathbf{log}(\boldsymbol{N} \,/\, \boldsymbol{n_k})$
- Sublinear function decreases the weight of "Rumpelstiltskin" compared to "book"
- Still monotonically increasing, so order is preserved

Reasoning about TF x IDF

- *Term frequency* high for common word in one document
- *Inverse Document Frequency* high for rare word in collection
- *Term-Frequency* x *Inverse-Document-Frequency* high for common word in one document that is rare in the collection
- $\boldsymbol{W_{ik} = tf_{ik} * \log(N / n_k)}$
  - T_k = term k in document D_i
  - tf_{ik} = freq of term T_k in doc D_i
  - N = total # docs in collection C
  - n_k = # docs in C that contain T_k

TF x IDF

- *Term-Frequency* x *Inverse-Document-Frequency*
  $\boldsymbol{W_{ik} = tf_{ik} * \log(N / n_k)}$
  - T_k = term k in document D_i
  - tf_{ik} = freq of term T_k in doc D_i
  - N = total # docs in collection *C*
  - n_k = # docs in *C* that contain T_k
- How would these affect the weight for a term T_k?
  - Large number of docs that contain T_k
  - Small number of docs that contain T_k
  - Large number of total documents
  - Small number of total documents

TF-IDF normalization

- Imagine two documents about kangaroos
- Document 1 mentions "kangaroo" 100 times. Total length 1000 words.
- Document 2 mentions "kangaroo" 200 times. Total length 10,000 words.

- Which document should have the greater weight?
  - Avoid giving longer doc more weight just because they are long
  - Need to normalize

TF-IDF normalization

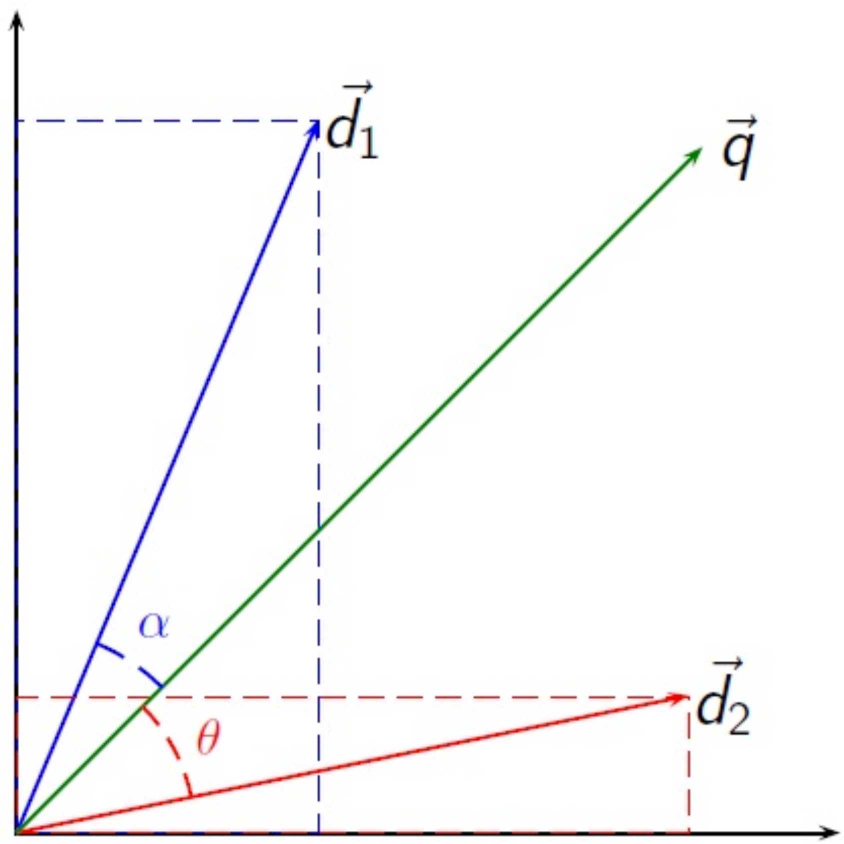
- Normalize term weights
  - Longer docs not given more weight
  - Normalize to sum-of-squares

$$w_{ik} = \frac{tf_{ik} \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log(N / n_k)]^2}}$$

- Some references use non-normalized tf-idf
  - $w_{ik} = tf_{ik} \log(N/n_k)$

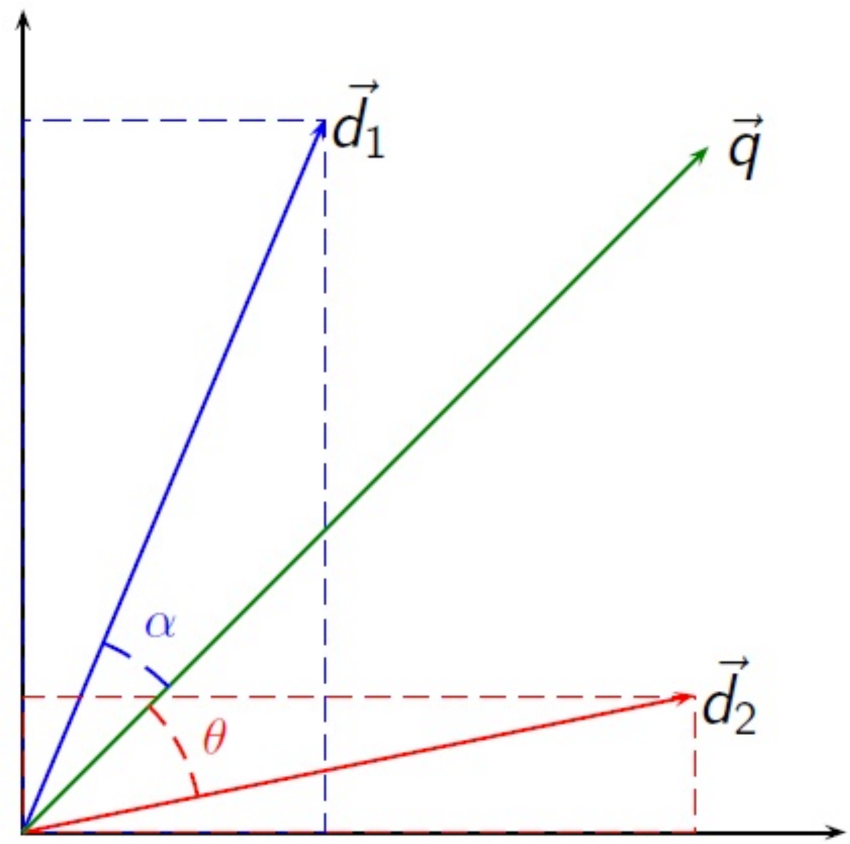
TF-IDF and the vector space

- Vector lengths are now TF-IDF values
- Vectors that are "close" are still similar in meaning



Vector space similarity

- Vector space similarity is also called the cosine, or normalized inner product
- Recall that cosine:
  - Depends on two adjacent vector lengths
  - =1 when angle is zero (points are identical)
  - Smaller when angle is greater



Vector space similarity

- Similarity of two docs is:

$$Sim(Di,Dj) = \sum_{k=1}^{t} w_{ik} * w_{jk}$$

Normalized ahead of time, when computing term weights.

$$\mathrm{sim}(d_j, q) = \frac{\mathbf{d_j} \cdot \mathbf{q}}{\|\mathbf{d_j}\| \, \|\mathbf{q}\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}}$$

Not normalized ahead of time

Vector space similarity

- Euclidean dot product formula

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Already "baked" into weight

$$Sim(Di,Dj) = \sum_{k=1}^{t} w_{ik} * w_{jk}$$

Vector space summary

- User's query treated as short document
- Query is in same space as docs
- Easy to measure a doc's distance to query
- Extension of Boolean retrieval

History

- IDF invented by Karen Spärck Jones
  - 1935 – 2007
- British Computer Scientist
- Proposed IDF in 1972 paper

Agenda

- What is search?
- Boolean retrieval
- Vector space model
- tf-idf
- **Assessing rank quality**
  - Precision/Recall Curves
  - Kendall’s Tau
  - Mean Reciprocal Rank

Assessing quality

- You've built a ranker. How do you know if it's any good?
- Relevance has been studied for a long time
  - Many contributing factors
  - People disagree on what is relevant
- Retrieval/assessment models differ
  - Binary relevance vs sorted relevance
  - Query-relevance vs user-relevance

Assessing quality

- Results from an experimental search engine
- BRITNEY IS BACK
- Query: "Britney"
- URL 1: http://www.britneyspears.com
- URL 2: http://andrewdeorio.com
- URL 3: http://en.wikipedia.org/wiki/Britney_Spears

Assessing quality

- Results from human "answer key"
- Query: "Britney"
- URL 1: http://www.britneyspears.com
- URL 2: http://en.wikipedia.org/wiki/Britney_Spears

  ...

- URL 90: http://andrewdeorio.com

Assessing quality

- Results from an experimental search engine
- Query: "Britney"
- URL 1: http://www.britneyspears.com
  - Human answer: 1
- URL 2: http://andrewdeorio.com
  - Human answer: 90
- URL 3: http://en.wikipedia.org/wiki/Britney_Spears
  - Human answer: 2

Assessing quality

- Results from an experimental search engine
  - Query: "Britney"
  - URLs: URL1, URL2, URL3, …
  - Rank: 1, 90, 2, ...
- Large hand-marked query/result tuples form the "answer key" for the ranker
- Text REtrieval Conference (TREC) is an annual conference, also publishes data
- Different tracks have included:
  - Blog track studies information-seeking
  - Chemical IR, Legal IR

Agenda

- What is search?
- Boolean retrieval
- Vector space model
- tf-idf
- Assessing rank quality
  - **Precision/Recall Curves**
  - Kendall's Tau
  - Mean Reciprocal Rank

Positives and negatives

- True positive
  - Relevant doc returned
- False positive
  - Irrelevant doc returned
- True negative
  - Irrelevant doc *not* returned
- False negative
  - Relevant doc *not* returned

Positives and negatives

- Label these docs as TP, FP, TN, FN
  - Query = puppies
- Search results
  - Britney Spears (@britneyspears) Instagram photos
  - 10 Dog Breeds That Have The CUTEST Puppies
- Web pages not included in search results
  - Cats - Reddit
  - Puppy Bowl XI Highlights

Positives and negatives

- Label these docs as TP, FP, TN, FN
  - Query = puppies
- Search results
  - Britney Spears (@britneyspears) Instagram photos **FP**
  - 10 Dog Breeds That Have The CUTEST Puppies **TP**
- Web pages not included in search results
  - Cats - Reddit **TN**
  - Puppy Bowl XI Highlights **FN**

Precision and recall

- Precision: fraction of retrieved docs that are relevant = relevant/retrieved
- Recall: fraction of relevant docs that are retrieved = retrieved/relevant

	Relevant	**Not Relevant**
Retrieved	TP	FP
Not retrieved	FN	TN

- Precision P = tp/(tp+fp)
- Recall R = tp/(tp+fn)

Precision and recall

- Generally trade precision vs. recall
  - How to get a system with high recall?
- Recall is a non-decreasing function of the # of docs retrieved
  - Precision **usually** decreases with more docs retrieved
- Drawbacks
  - Binary relevance
  - Need human judgments
  - Must average over large corpus
  - Alternatively, skewed by corpus/author selection

Precision: % of selected items that are correct

Recall: % of correct items that are selected

Exercise

- A search engine always returns all documents

- Do you expect high or low precision?

- Do you expect high or low recall?

Exercise

Precision: % of selected items that are correct

Recall: % of correct items that are selected

- A search engine always returns all documents

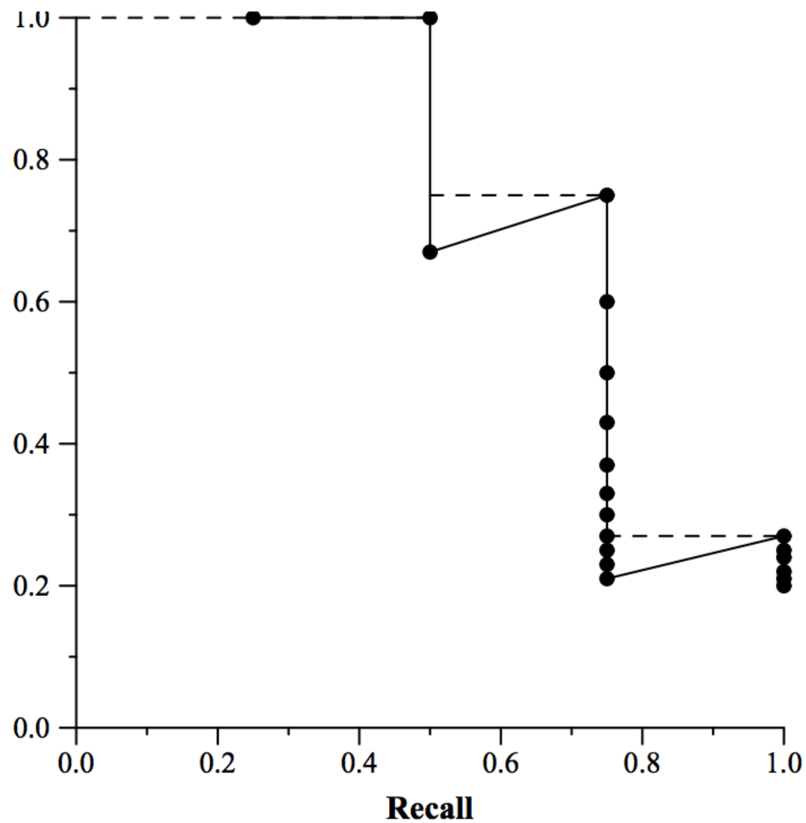
- Do you expect high or low precision?
  - Low. If all docs are returned, then many non-relevant docs are included, which will decrease the percentage of returned docs that are relevant.
- Do you expect high or low recall?
  - High. If all docs are returned, then all relevant docs must be returned.

Precision-recall curves

- A search engine will create a total ordering on all documents
- The top k are returned to the user
- We can calculate precision and recall for several values of k
- This creates a *precision-recall curve*

Precision-recall example

- Collection of 20 documents
- Relevant docs are ranked 1, 2, 4, 15



Precision: % of selected items that are correct

Recall: % of correct items that are selected

http://trec.nist.gov/pubs/trec15/appendices/CE.MEASURES06.pdf

Agenda

- What is search?
- Boolean retrieval
- Vector space model
- tf-idf
- Assessing rank quality
  - Precision/Recall Curves
  - **Kendall's Tau**
  - Mean Reciprocal Rank

Take ranking into account

- Precision at fixed recall
  - Precision of top k results, for k=1,10,50,…
  - Critical for Web Search
- Kendall's Tau for comparing sorts

Kendall's tau

- Use a real ordering of documents, not just binary "relevant/not relevant"
- The correct document ordering is:
  - 1, 2, 3, 4
- Search Engine A outputs:
  - 1, 2, 4, 3
- Search Engine B outputs:
  - 4, 3, 1, 2
- Intuitively, A is better. How do we capture this numerically?

Measuring rank correlation

- Kendall's Tau has some nice properties:
  - If agreement between 2 ranks is perfect, then KT = 1
  - If disagreement is perfect,
    then KT = -1
  - If rankings are uncorrelated, then KT = 0 on average
- Intuition: Compute fraction of pairwise orderings that are consistent

Kendall's tau

pairs that agree

pairs that disagree

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

total # pairs

- The non-normalized version is called Kendall's Tau Distance
- Also called *bubble-sort distance*

Kendall's tau example

- Correct ordering:
  - 1, 2, 3, 4
- Search Engine A:
  - 1, 2, 4, 3

$$\tau = \frac{5-1}{\frac{1}{2}4(4-1)} = \frac{4}{6} = 0.666$$

- Search Engine B:
  - 4, 3, 2, 1

$$\tau = \frac{0-6}{\frac{1}{2}4(4-1)} = \frac{-6}{6} = -1$$

Agenda

- What is search?
- Boolean retrieval
- Vector space model
- tf-idf
- Assessing rank quality
  - Precision/Recall Curves
  - Kendall's Tau
  - **Mean Reciprocal Rank**

Mean reciprocal rank

- "How close to the top of the search results is the 1st correct answer?"

$$meanreciprocalrank = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

- Good for search results
- Great for systems that return a single guess

Mean reciprocal rank example

Query set "Q"
- web.eecs.umich.edu/~jflinn/
- andrewdeorio.com
- **en.wikipedia.org/wiki/Britney_Spears**

Correct answer
- britneyspears.com
- instagram.com/britneyspears
- en.wikipedia.org/wiki/Britney_Spears

$$meanreciprocalrank = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

Query: "britney"
MRR = 1/3

Mean reciprocal rank

$$meanreciprocalrank = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

Windy City?	Toronto, Chicago, NYC
Tree City?	Ann Arbor, Madison, Capital City
Emerald City?	Vancouver, San Francisco, Seattle
MRR	

Mean reciprocal rank

$$meanreciprocalrank = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

Query	**Results**	**Rank**	**Reciprocal rank**
Windy City?	Toronto, Chicago, NYC	2	1/2
Tree City?	Ann Arbor, Madison, Capital City	1	1
Emerald City?	Vancouver, San Francisco, Seattle	3	1/3
MRR			**0.611**

Assessing rank quality

- Precision and Recall
  - Usually trade off each other
    - Precision-recall curve
  - Requires relevant/not-relevant judgments
- Kendall's Tau
  - Measures correlation between two rankings
    - "Fraction of pairs in agreement"
  - +1 if perfect agreement; -1 disagreement
- Mean Reciprocal Rank
  - "How close to the top of the search results is the 1st correct answer?"

Summary

- Today we used the words on a page to rank search results
- Next time, we'll use the links between web pages to improve search results