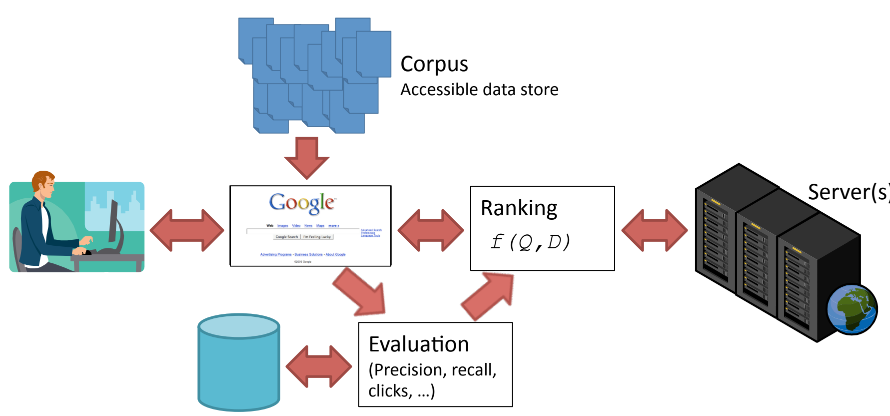
Relevance Feedback

CISC489/689-010, Lecture #15

Monday, April 13th

Ben Carterette

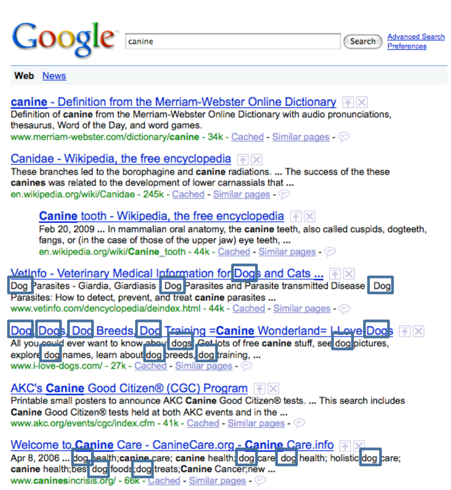
Query Process

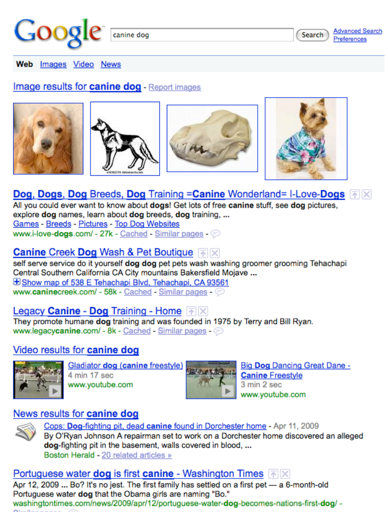


User Interaction

- User inputs a query
- Gets a ranked list of results
- Interaction doesn't have to end there!
  - A typical engine-user interaction: the user looks at the results and reformulates the query
  - What if the engine could do it automatically?

Example

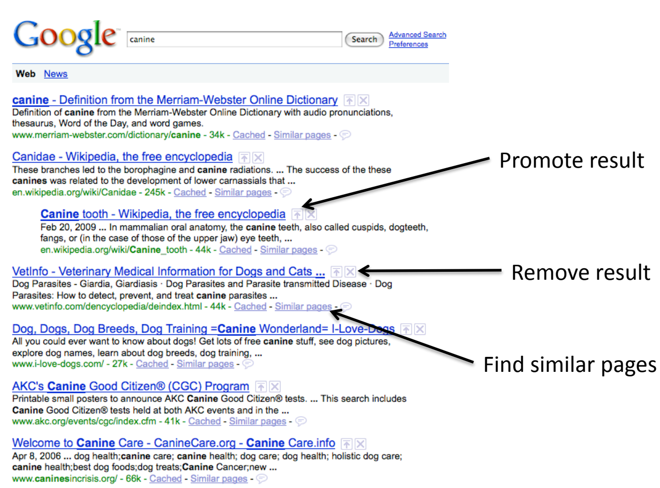




Interaction Model

- *Relevance feedback*
  - User indicates which documents were relevant, which were nonrelevant
    - Possibly using check boxes or some other button
  - System takes this *feedback* and uses it to find other relevant documents
  - Typical approach: *query expansion*
  - Add "relevant terms" to the query with weights

Example Feedback Interface



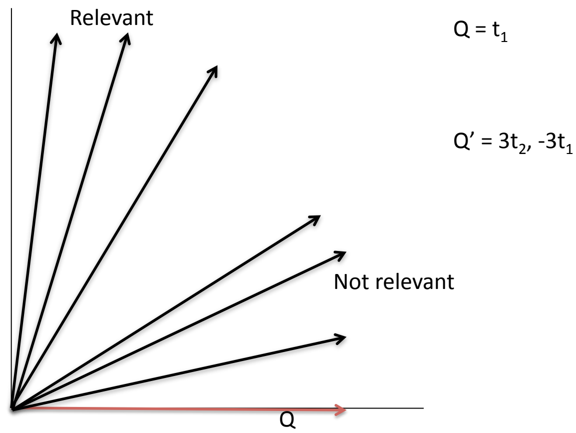
Models for Relevance Feedback

- Retrieval models <-> relevance feedback models
- A model for relevance feedback needs to take marked relevant documents and use them to update the query or results
  - Google model is very simple: move result to top on "promote" click, move to bottom on "remove" click
  - Slightly more complex Google model: use one document as a relevant document for "similar pages" click
  - Query expansion is a more common approach

Vector Space Feedback

- Documents, queries are vectors
- Add relevant document vectors together to obtain a "relevant vector"
- Add nonrelevant document vectors together to obtain a "nonrelevant vector"
- We want a new query vector Q' that is closer to the relevant vector than the nonrelevant vector

VSM Feedback Illustration



Relevance Feedback

- Rocchio algorithm
- *Optimal query*
  - Maximizes the difference between the average vector representing the relevant documents and the average vector representing the non-relevant documents
- Modifies query according to

$$q'_j = \alpha . q_j + \beta . \frac{1}{|Rel|} \sum_{D_i \in Rel} d_{ij} - \gamma . \frac{1}{|Nonrel|} \sum_{D_i \in Nonrel} d_{ij}$$

  - α, β, and γ are parameters
    - Typical values 8, 16, 4

Rocchio Feedback in Practice

$$q'_j = \alpha . q_j + \beta . \frac{1}{|Rel|} \sum_{D_i \in Rel} d_{ij} - \gamma . \frac{1}{|Nonrel|} \sum_{D_i \in Nonrel} d_{ij}$$

- Might add top *k* terms only
- Could ignore the nonrelevant part
  - Has not consistently been shown to improve performance
- Might choose to include some documents but not others
  - Most certain, most uncertain, highest quality, …

Rocchio Expanded Query Example

- TREC topic 106:

  Title: U.S. Control of Insider Trading
  Description: Document will report proposed or enacted changes to U.S. laws and regulations designed to prevent insider trading.

- Original query (automatically generated):

```
#wsum( 2.0 #uw50( Control of Insider Trading )
       2.0 #1( #USA Control )
       5.0 #1( Insider Trading )
       1.0 proposed 1.0 enacted 1.0 changes 1.0 #1( #USA laws )
       1.0 regulations 1.0 designed 1.0 prevent )
```

- Expanded query:

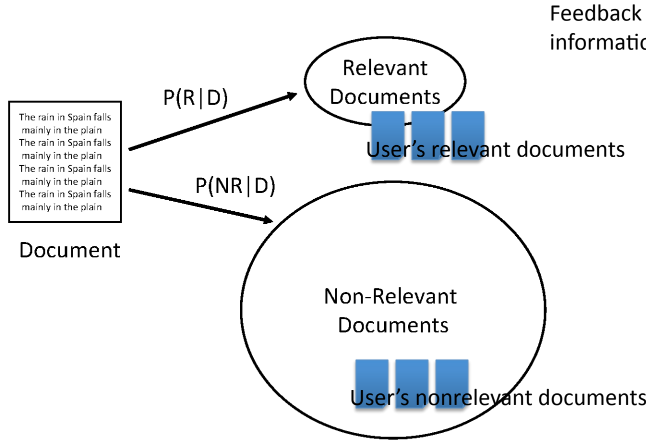
```
#wsum( 3.88 #uw50( control inside trade ) 2.21 #1( #USA control )
       145.57 #1( inside trade )
       0.54 propose 2.46 enact 0.99 change 4.35 #1( #USA law )
       10.35 regulate 0.80 design 1.73 prevent
       4.60 drexel 2.05 fine 1.85 subcommittee 1.69 surveillance 1.60 markey
       1.53 senate 1.19 manipulate 1.10 pass 1.06 scandal 0.92 edward )
```

Probabilistic Feedback

- Recall probabilistic models:
  - Relevant class versus nonrelevant class
    - P(R | D, Q) versus P(NR | D, Q)
  - Optimal ranking is in decreasing order of probability of relevance
- Basic probabilistic model assumes no knowledge of classes
  - e.g. BIM: $\log \frac{0.5(1-\frac{n_i}{N})}{\frac{n_i}{N}(1-0.5)} = \log \frac{N-n_i}{n_i}$

Illustration

Feedback provides information about the classes



Contingency Table

For term i:

	Relevant	Non-relevant	Total
$d_i = 1$	$\boxed{r_i}$	$\boxed{n_i} - r_i$	n_i
$d_i = 0$	$R - r_i$	$N - n_i - R + r_i$	$N - r_i$
Total	$\boxed{R}$	$\boxed{N} - R$	N

Number of relevant documents that contain term i

Number of relevant documents

Number of documents

Number of documents that contain term i

$$p_i = (r_i + 0.5)/(R + 1)$$

$$s_i = (n_i - r_i + 0.5)/(N - R + 1)$$

Gives BIM feedback scoring function:

$$\sum_{i:d_i=q_i=1} \log \frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}$$

BIM Feedback

- Not query expansion
  - It does not add terms to the query
- It modifies term weights based on presence or absence in relevant documents
  - Terms that appear much more often in the relevant class than the nonrelevant class are good discriminators of relevance
  - i.e. $r_i > n_i - r_i$ → good discriminator

Language Model Feedback

- Recall the query-likelihood language model:
  $$P(Q|D) = \prod_{t \in Q} P(t|D)$$
  - Where's the relevance?
- A *relevance model* is a language model for the information need
  - P(t | R)
  - What is the probability that the author of some relevant document would use the term *t*?
  - Or what is the probability that the user with the information need would describe it using *t*?

Relevance Models

- The query and relevant documents are samples from the relevance model
- *P(D|R)* - probability of generating the text in a document given a relevance model
  - *document likelihood* model
  - less effective than query likelihood due to difficulties comparing across documents of different lengths
- Original motivation was to incorporate relevance into language model

Estimating the Relevance Model

- Probability of pulling a word *w* out of the "bucket" representing the relevance model depends on the *n* query words we have just pulled out

$$P(w|R) \approx P(w|q_1 \ldots q_n)$$

- By definition

$$P(w|R) \approx \frac{P(w, q_1 \ldots q_n)}{P(q_1 \ldots q_n)}$$

Estimating the Relevance Model

- Joint probability is

$$P(w, q_1 \ldots q_n) = \sum_{D \in \mathcal{C}} p(D) P(w, q_1 \ldots q_n | D)$$

- Assume

$$P(w, q_1 \ldots q_n | D) = P(w|D) \prod_{i=1}^{n} P(q_i|D)$$

- Gives

$$P(w, q_1 \ldots q_n) = \sum_{D \in \mathcal{C}} P(D) P(w|D) \boxed{\prod_{i=1}^{n} P(q_i|D)}$$

Look familiar?

Query-likelihood score. Set to 0 for nonrelevant docs.

Estimating the Relevance Model

- *P(D)* usually assumed to be uniform
- *P(w, q1 . . . qn)* is simply a weighted average of the language model probabilities for *w* in a set of documents, where the weights are the query likelihood scores for those documents
- Formal model for relevance feedback in the language model
  - query expansion technique

Relevance Models in Practice

- In theory:
  - Use all the documents in the collection weighted by query-likelihood score or relevance
  - Expand query with every term in the vocabulary
- In practice:
  - Use only the feedback documents, or the top *k* documents, or a subset
  - Expand query with only *n* highest-probability terms

Example RMs from Top 10 Docs

president lincoln	*abraham lincoln*	*fishing*	*tropical fish*
lincoln	lincoln	fish	fish
president	america	farm	tropic
room	president	salmon	japan
bedroom	faith	new	aquarium
house	guest	wild	water
white	abraham	water	species
america	new	caught	aquatic
guest	room	catch	fair
serve	christian	tag	china
bed	history	time	coral
washington	public	eat	source
old	bedroom	raise	tank
office	war	city	reef
war	politics	people	animal
long	old	fishermen	tarpon
abraham	national	boat	fishery

Example RMs from Top 50 Docs

president lincoln	*abraham lincoln*	*fishing*	*tropical fish*
lincoln	lincoln	fish	fish
president	president	water	tropic
america	america	catch	water
new	abraham	reef	storm
national	war	fishermen	species
great	man	river	boat
white	civil	new	sea
war	new	year	river
washington	history	time	country
clinton	two	bass	tuna
house	room	boat	world
history	booth	world	million
time	time	farm	state
center	politics	angle	time
kennedy	public	fly	japan
room	guest	trout	mile

KL-Divergence

- Given the *true* probability distribution *P* and another distribution *Q* that is an *approximation* to *P*,

$$KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

  - Use negative KL-divergence for ranking, and assume relevance model *R* is the true distribution (not symmetric),

Scoring function

$$\boxed{\sum_{w \in V} \underbrace{\boxed{P(w|R)}}_{\text{Relevance model}} \log \underbrace{\boxed{P(w|D)}}_{\text{Document language model}} - \sum_{w \in V} P(w|R) \log P(w|R)}$$

KL-Divergence

- Given a simple maximum likelihood estimate for *P(w|R),* based on the frequency in the query text, ranking score is

$$\sum_{w \in V} \frac{f_{w,Q}}{|Q|} \log P(w|D)$$

  - rank-equivalent to query likelihood score
- Query likelihood model is a special case of retrieval based on relevance model

Language Model Feedback: Another Perspective

- Language model uses *smoothing*:

$$P(Q|D) = \prod_{t \in Q} P(t|D) = \prod_{t \in Q} \alpha_D \frac{tf_{t,D}}{|D|} + (1 \quad \alpha_D) \frac{ctf_t}{|C|}$$

- Smoothing "expands" the document with terms that were not originally included
- *Document expansion*
  - Instead of modifying query representation, modify document representation
- Language model performs expansion by default

Testing Relevance Feedback

- Let's say we implement relevance feedback
  - Our index allows us to find all of the terms contained in a document
  - The interface allows the user to specify "relevant" or "not relevant" for each document
  - We have implemented some query expansion method like Rocchio
- How do we determine whether it's useful?

Testing Relevance Feedback

- System-based measures (precision, recall, etc) can tell us whether relevance feedback is *effective*
- User studies can tell us whether users actually like it or not

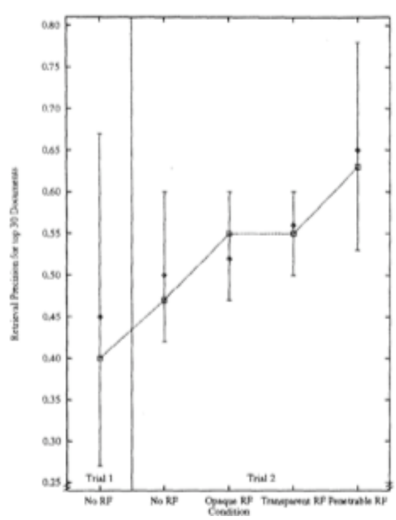
A User Study

- Koenemann and Belkin, "A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness", CHI 1996
- User study with 64 subjects
- Three different types of feedback:
  – System does pseudo-feedback without user's knowledge ("opaque")
  – System does pseudo-feedback and shows expanded query to user ("transparent")
  – System does pseudo-feedback but allows user to modify expanded query before reranking ("penetrable")

Experimental Procedure

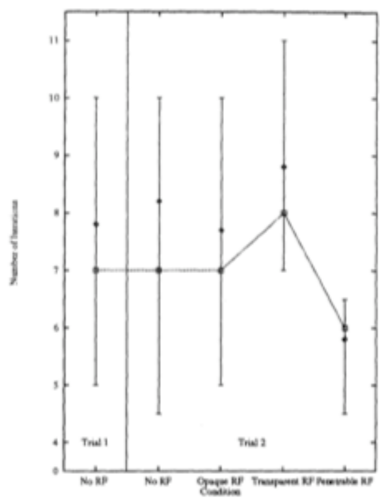
- Users submit a query
  - First without relevance feedback
  - Second based on one of three feedback approaches (selected randomly)
- System evaluation based on last query submitted
- With no RF, no difference between users

Effectiveness With Feedback



- RF gives clear improvement
- “Opaque” and “transparent” same effectiveness
- “Penetrable” best

Number of Queries



- How many queries did users try before stopping?
- "Transparent" resulted in one additional query
- "Penetrable" resulted in one fewer

Feedback Uptake

Mean Number & Sources of Query Terms					
Relevance Feedback Condition	User Controlled			Added by RF SYS	Σ
	User Typed	Copy from RF	Σ		
Topic 162:					
None	6.9	n/a	6.9	n/a	6.9
Opaque	10.9	n/a	10.9	35.9	46.8
Transparent	3.3	9.1	12.4	42.8	55.1
Penetrable	6.3	24.4	30.6	n/a	30.6
Topic 165:					
None	6.0	n/a	6.0	n/a	6.0
Opaque	3.8	n/a	3.8	20.5	24.3
Transparent	4.3	5.3	9.5	17.8	27.3
Penetrable	3.3	9.5	12.8	n/a	12.8
162&165:					
None	6.4	n/a	6.4	n/a	6.4
Opaque	7.3	n/a	7.3	28.2	35.5
Transparent	3.8	7.2	10.9	30.3	41.2
Penetrable	4.8	16.9	21.7	n/a	21.7

- Users used short queries
- But they often "copied" words from the expanded terms
- Shorter queries with more transparent feedback

User Reactions

- Subjects liked being able to see and select feedback terms ("penetrable")
- Those in the "opaque" setting wanted to be able to see what was happening
- Subjects used feedback to put less effort into formulating queries, instead putting effort into choosing terms

Pseudo-Relevance Feedback

- Instead of making the user give feedback, let's just assume the top documents are relevant
- Use those to expand the query
- Re-rank documents with new query, show only the final results to the user

Pseudo-Feedback Algorithm for RM

1. Rank documents using the query likelihood score for query Q.
2. Select some number of the top-ranked documents to be the set $\mathcal{C}$.
3. Calculate the relevance model probabilities $P(w|R)$. $P(q_1 \ldots q_n)$ is used as a normalizing constant and is calculated as

$$P(q_1 \ldots q_n) = \sum_{w \in V} P(w, q_1 \ldots q_n)$$

4. Rank documents again using the KL-divergence score

$$\sum_{w} P(w|R) \log P(w|D)$$

Testing Psuedo-Relevance Feedback

- Does it work?
  - Effectiveness measures only; user does not need to be involved
- Common result at TREC:
  - Small but statistically significant improvement in mean average precision
    - e.g. Rocchio improved MAP from 0.373 to 0.407 at TREC in 1993
    - Relevance models improve MAP significantly at recent TRECs
  - Some queries improve, some get much worse

Experimental Results

50 TREC topics (numbers 401-450)
Evaluated with mean average precision