Relevance Feedback and Query Expansion

Chapter 9 - IIR

SAPIENZA
UNIVERSITÀ DI ROMA

Fabrizio Silvestri

Quick Recap

Relevance

- We will evaluate the quality of an information retrieval system and, in particular, its ranking algorithm with respect to relevance.
- A document is relevant if it gives the user the information she was looking for.
- To evaluate relevance, we need an evaluation benchmark with three elements:
  - A benchmark document collection
  - A benchmark suite of queries
  - An assessment of the relevance of each query-document pair

Query vs. Information Need

- The notion of “relevance to the query” is very problematic.
- Information need i: You are looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.
- Query q: wine and red and white and heart and attack
- Consider document d′:
  - He then launched into the heart of his speech and attacked the wine industry lobby for downplaying the role of red and white wine in drunk driving.
- d′ is relevant to the query q, but d′ is **not** relevant to the information need i.
- User happiness/satisfaction (i.e., how well our ranking algorithm works) can only be measured by relevance to information needs, not by relevance to queries.

Precision and Recall

- Precision (P) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

- Recall (R) is the fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

Motivation

Improving Recall

- Main topic today: two ways of improving recall: relevance feedback and query expansion
- As an example consider query q: [aircraft] . . .
- . . . and document d containing “plane”, but not containing “aircraft”
- A simple IR system will not return d for q.
- Even if d is the most relevant document for q!
- We want to change this:
    - Return relevant documents even if there is no term match with the (original) query

In fact… a lousier definition of Recall

- Loose definition of recall in this lecture: “increasing the number of relevant documents returned to user”
- This may actually decrease recall on some measures, e.g., when expanding “jaguar” to “jaguar AND panthera”
    - . . . which eliminates some relevant documents, but increases relevant documents returned on top pages

Options for improving recall

- Local: Do a “local”, on-demand analysis for a user query Main local method: relevance feedback
  - Part 1
- Global: Do a global analysis once (e.g., of collection) to produce thesaurus Use thesaurus for query expansion
  - Part 2

Options for improving recall

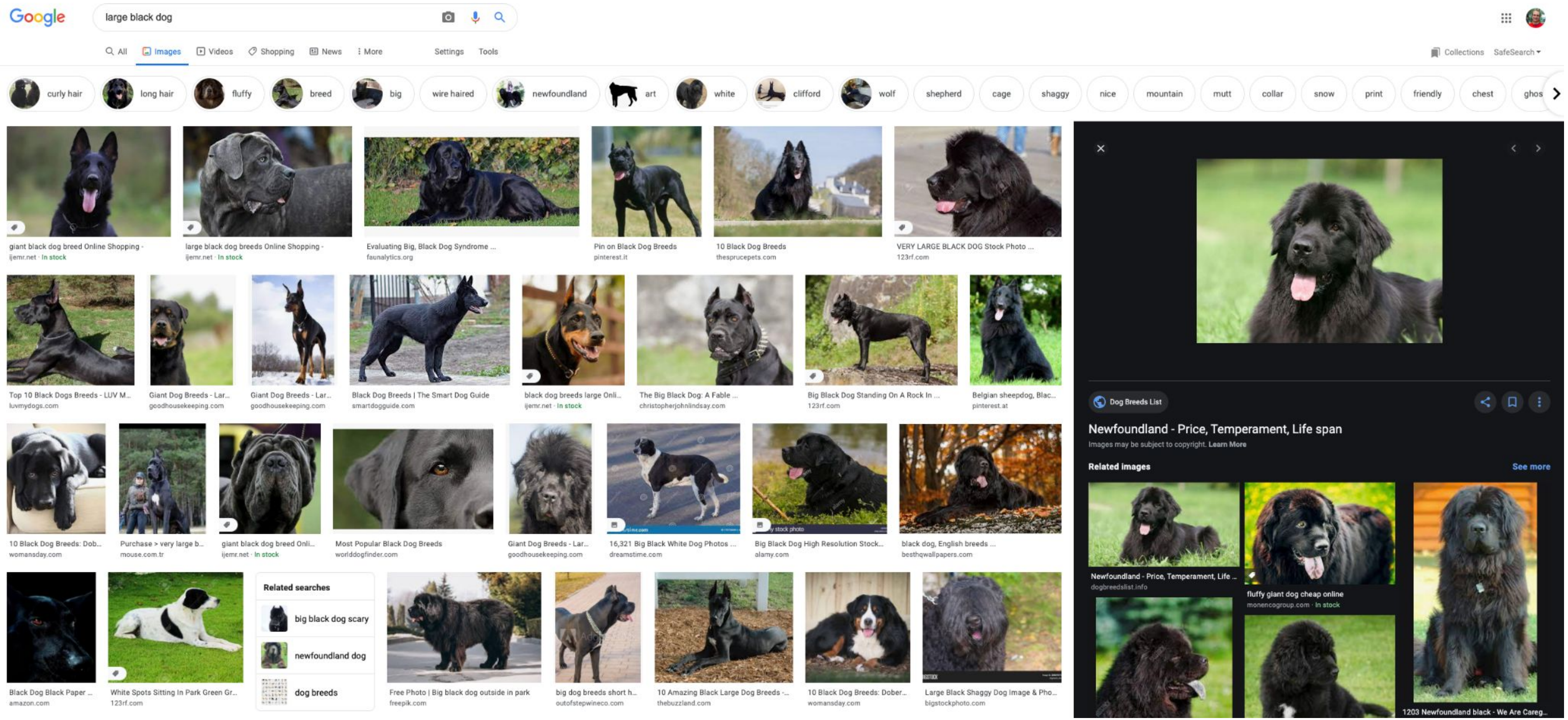
- Local: Do a “local”, on-demand analysis for a user query Main local method: relevance feedback
  - Part 1
- Global: Do a global analysis once (e.g., of collection) to produce thesaurus Use thesaurus for query expansion
  - Part 2

Relevance Feedback: Basics

Relevance feedback: Basic idea

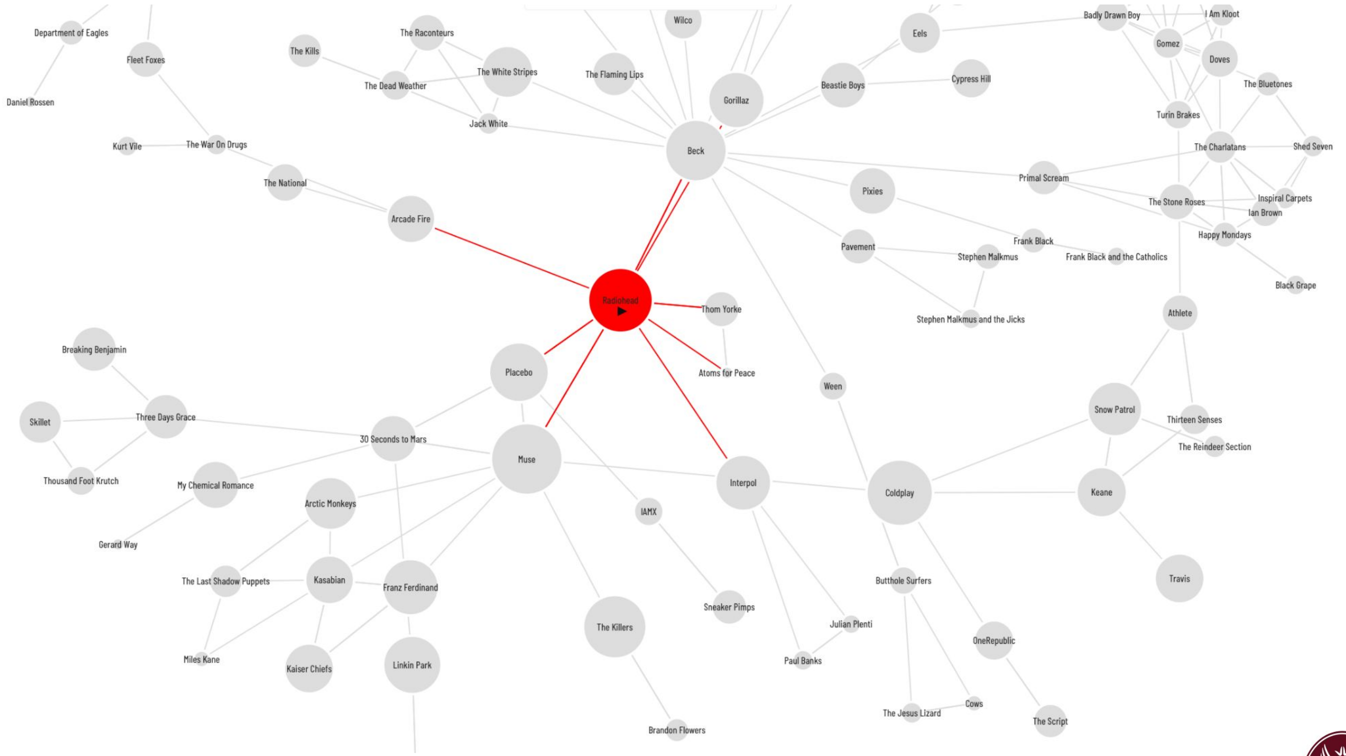
- The user issues a (short, simple) query.
- The search engine returns a set of documents.
- User marks some docs as relevant, some as nonrelevant.
- Search engine computes a new representation of the information need. Hope: better than the initial query.
- Search engine runs new query and returns new results.
- New results have (hopefully) better recall.
- We will use the term ad hoc retrieval to refer to regular retrieval without relevance feedback.

Relevance feedback: Example

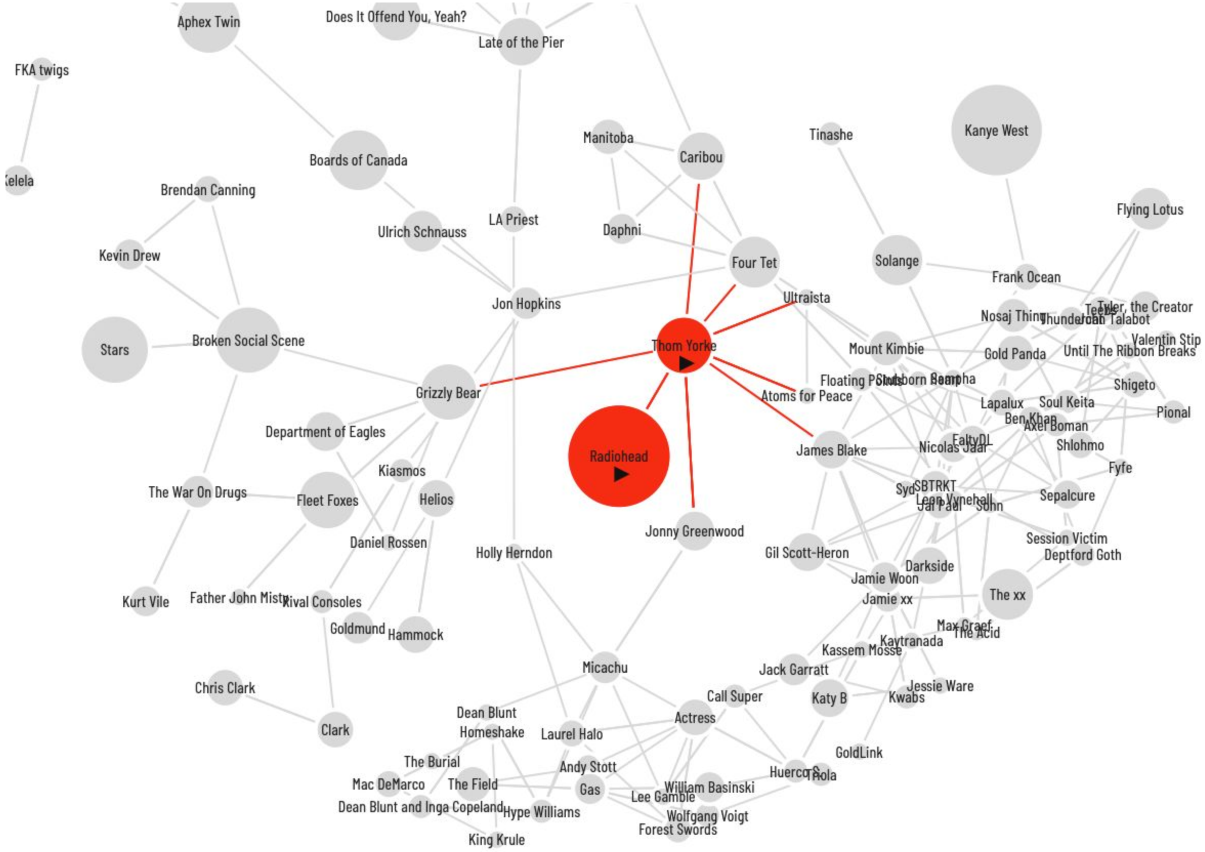


SAPIENZA
UNIVERSITÀ DI ROMA

Relevance feedback: Liveplasma.com



Relevance feedback: Liveplasma.com



SAPIENZA
UNIVERSITÀ DI ROMA

Relevance Feedback: Details

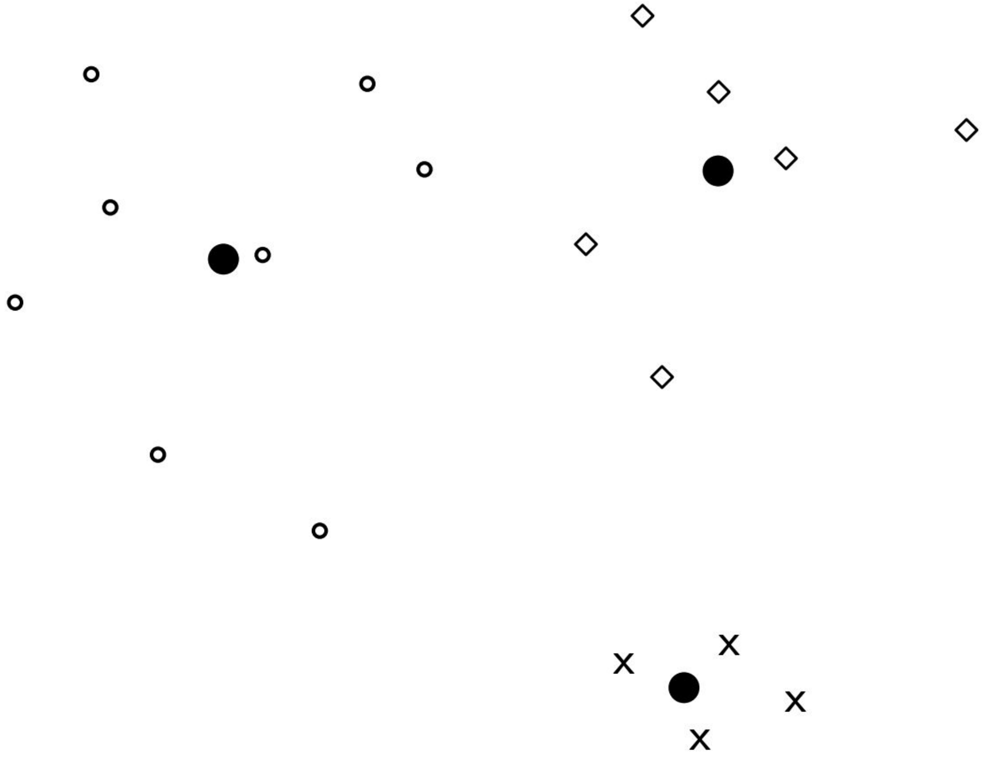
Key Concepts: The Centroid

- The centroid is the center of mass of a set of points.
- Recall that we represent documents as points in a high-dimensional space.
- Thus: we can compute centroids of documents.
- Definition:

$$\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$$

- where D is a set of documents and v(d) is the vector we use to represent document d.

Centroids



Rocchio

- The Rocchio algorithm implements relevance feedback in the vector space model.
- Rocchio chooses the query q_{opt} that maximizes

$$\vec{q}_{opt} = \arg\max_{\vec{q}} [\text{sim}(\vec{q}, \mu(D_r)) - \text{sim}(\vec{q}, \mu(D_{nr}))]$$

- D_r: set of relevant docs; D_{nr}: set of nonrelevant docs
- Intent: q_{opt} is the vector that separates relevant and nonrelevant docs maximally.
- Making some additional assumptions (sim is cosine similarity0), we can rewrite q_{opt} as: $\vec{q}_{opt} = \mu(D_r) + [\mu(D_r) - \mu(D_{nr})]$

SAPIENZA UNIVERSITÀ DI ROMA

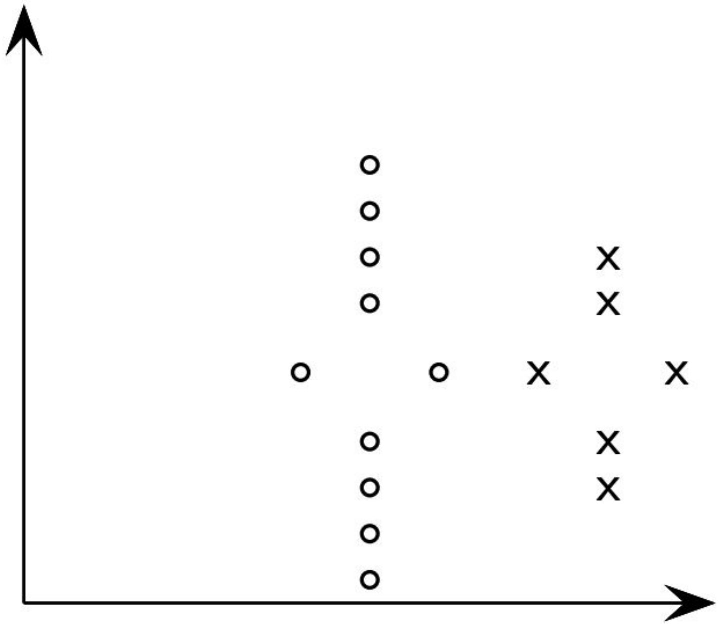
Optimal Query Vector

- The optimal query vector is:

$$
\begin{aligned}
\vec{q}_{opt} &= \mu(D_r) + [\mu(D_r) - \mu(D_{nr})] \\
&= \frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r} \vec{d}_j + [\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}} \vec{d}_j]
\end{aligned}
$$

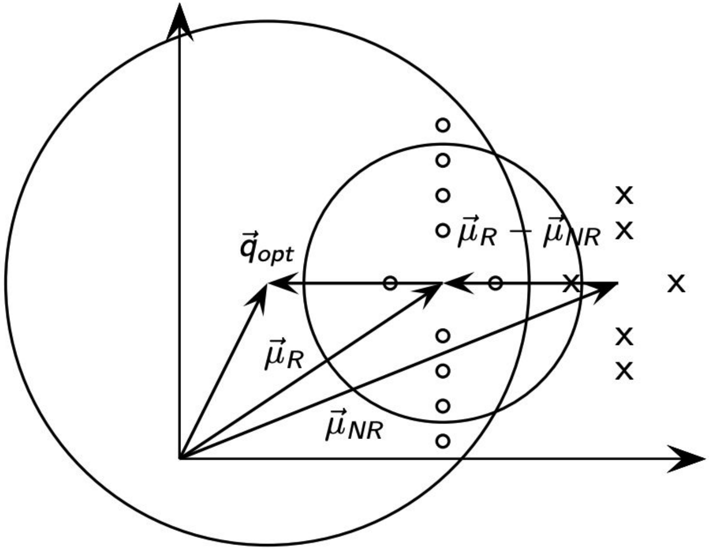
- We move the centroid of the relevant documents by the difference between the two centroids.

Example: Compute Rocchio Vector



- O's relevant docs, X's nonrelevant docs
- Compute $\vec{q}_{opt} \quad = \quad \mu(D_r) + [\mu(D_r) - \mu(D_{nr})]$

Example: Compute Rocchio Vector



- Os: relevant documents, Xs: nonrelevant documents
- μ(R): centroid of relevant documents μ(R) does not separate relevant/nonrelevant. μ(NR): centroid of nonrelevant documents; μ(R) − μ(NR): difference vector
- Add difference vector to μ(R) to get q_{opt}, which separates relevant/nonrelevant perfectly.

SAPIENZA
UNIVERSITÀ DI ROMA

Rocchio as it is usually implemented

- Used in practice:

$$\begin{aligned}\vec{q}_m &= \alpha\vec{q}_0 + \beta\mu(D_r) - \gamma\mu(D_{nr}) \\ &= \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j\in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j\in D_{nr}}\vec{d}_j\end{aligned}$$

  q_m: modified query vector; q_0: original query vector; D_r and D_{nr}: sets of known relevant and nonrelevant documents respectively; α, β, and γ: weights
- New query moves towards relevant documents and away from nonrelevant documents.
- Tradeoff α vs. β/γ: If we have a lot of judged documents, we want a higher β/γ.
- Set negative term weights to 0.
- “Negative weight” for a term doesn’t make sense in the vector space model.

Rocchio as it is usually implemented

- Used in practice:

$$\begin{aligned}\vec{q}_m &= \alpha\vec{q}_0 + \beta\mu(D_r) - \gamma\mu(D_{nr})\\ &= \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j\in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j\in D_{nr}}\vec{d}_j\end{aligned}$$

  q_m: modified query vector; q_0: original query vector; D_r and D_{nr}: sets of known relevant and nonrelevant documents respectively; α, β, and γ: weights
- New query moves towards relevant documents and away from nonrelevant documents.
- Tradeoff α vs. β/γ: If we have a lot of judged documents, we want a higher β/γ.
- Set negative term weights to 0.
- "Negative weight" for a term doesn't make sense in the vector space model.

Positive vs. Negative Feedback

- Positive feedback is more valuable than negative feedback.
- For example, set $\beta = 0.75$, $\gamma = 0.25$ to give higher weight to positive feedback.
- Many systems only allow positive feedback.

Relevance Feedback: Assumptions

- When can relevance feedback enhance recall?
- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Assumption A2: Relevant documents contain similar terms (so I can “hop” from one relevant document to a different one when giving relevance feedback).

Violations of A1

- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Violation: Mismatch of searcher’s vocabulary and collection vocabulary
- Example: cosmonaut / astronaut

Violations of A2

- Assumption A2: Relevant documents are similar.
- Example for violation: [contradictory government policies]
- Several unrelated "prototypes"
    - Subsidies for tobacco farmers vs. anti-smoking campaigns
    - Aid for developing countries vs. high tariffs on imports from developing countries
- Relevance feedback on tobacco docs will not help with finding docs on developing countries.

Evaluating Relevance Feedback

- Pick an evaluation measure, e.g., precision in top 10: P@10
- Compute P@10 for original query q0
- Compute P@10 for modified relevance feedback query q1
- In most cases: q1 is spectacularly better than q0!
- Is this a fair evaluation?

Evaluating Relevance Feedback

- Fair evaluation must be on “residual” collection: docs not yet judged by user.
- Studies have shown that relevance feedback is successful when evaluated this way.
- Empirically, one round of relevance feedback is often very useful. Two rounds are marginally useful.

Evaluating Relevance: Caveat Emptor

- True evaluation of usefulness must compare to other methods taking the same amount of time.
- Alternative to relevance feedback: User revises and resubmits query.
- Users may prefer revision/resubmission to having to judge relevance of documents.
- There is no clear evidence that relevance feedback is the "best use" of the user's time.

Issues with Relevance Feedback

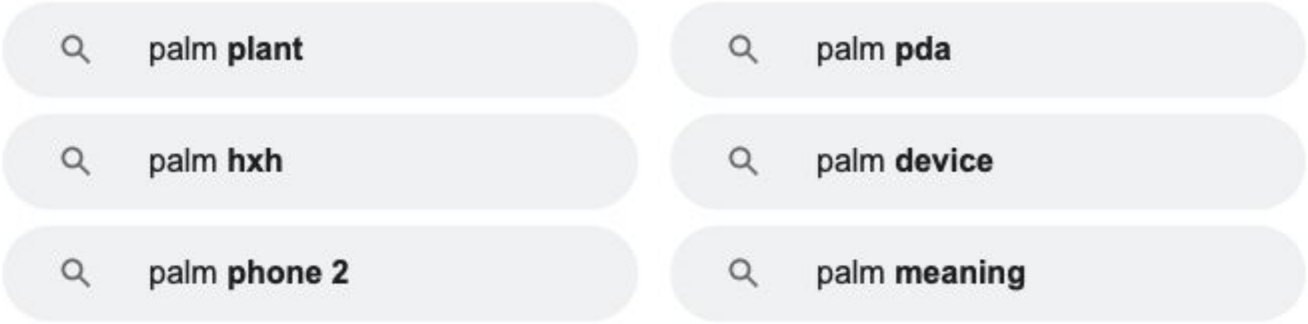
- Relevance feedback is expensive.
- Relevance feedback creates long modified queries.
- Long queries are expensive to process.
- Users are reluctant to provide explicit feedback.
- It’s often hard to understand why a particular document was retrieved after applying relevance feedback.

Pseudo Relevance Feedback

- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.
- Pseudo-relevance feedback algorithm:
  - Retrieve a ranked list of hits for the user’s query
  - Assume that the top k documents are relevant.
  - Do relevance feedback (e.g., Rocchio)
- Works very well on average
- But can go horribly wrong for some queries.
- Because of query drift
  - If you do several iterations of pseudo-relevance feedback, then
  - you will get query drift for a large proportion of queries.

Query Expansion

What is Query Expansion



SAPIENZA
UNIVERSITÀ DI ROMA

What is Query Expansion

- Query expansion is another method for increasing recall.
- We use “global query expansion” to refer to “global methods for query reformulation”.
- In global query expansion, the query is modified based on some global resource, i.e. a resource that is not query-dependent.
- Main information we use: (near-)synonymy

“Global” resources used for query expansion

- A publication or database that collects (near-)synonyms is called a thesaurus.
- Manual thesaurus (maintained by editors, e.g., PubMed)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics)
- Query-equivalence based on query log mining (common on the web as in the “palm” example)

Thesaurus-based Query Expansion

- For each term t in the query, expand the query with words the thesaurus lists as semantically related with t.
- Example from earlier: hospital → medical
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
  - interest rate → interest rate fascinate
- Widely used in specialized search engines for science and engineering
- It’s very expensive to create a manual thesaurus and to maintain it over time.

Automatic Thesaurus Generation

- Attempt to generate a thesaurus automatically by analyzing the distribution of words in documents
- Fundamental notion: similarity between two words
- Definition 1: Two words are similar if they co-occur with similar words.
  - “car” ≈ “motorcycle” because both occur with “road”, “gas” and “license”, so they must be similar.
- Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.
  - You can harvest, peel, eat, prepare, etc. apples and pears, so apples and pears must be similar.
- Co-occurrence is more robust, grammatical relations are more accurate.

Co-occurence-based thesaurus: Examples

Word	Nearest neighbors
absolutely	absurd whatsoever totally exactly nothing
bottomed	dip copper drops topped slide trimmed
captivating	shimmer stunningly superbly plucky witty
doghouse	dog porch crawling beside downstairs
makeup	repellent lotion glossy sunscreen skin gel
mediating	reconciliation negotiate case conciliation
keeping	hoping bring wiping could some would
lithographs	drawings Picasso Dali sculptures Gauguin
pathogens	toxins bacteria organisms bacterial parasite
senses	grasp psyche truly clumsy naive innate

Query Expansion at Search Engines

- Main source of query expansion at search engines: query logs
- Example 1: After issuing the query [herbs], users frequently search for [herbal remedies].
    - → “herbal remedies” is potential expansion of “herb”.
- Example 2: Users searching for [flower pix] frequently click on the URL photobucket.com/flower. Users searching for [flower clipart] frequently click on the same URL.
    - → “flower clipart” and “flower pix” are potential expansions of each other.

Main Takeaways

- Interactive relevance feedback: improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant
- Best known relevance feedback method: Rocchio feedback
- Query expansion: improve retrieval results by adding synonyms / related terms to the query
- Sources for related terms: Manual thesauri, automatic thesauri, query logs