Relevance feedback and query expansion: tuning the query

Stanislav Protasov

Agenda

- What if results for the query are not satisfactory?
  - Local methods of improvement
  - Global methods
- How to suggest continuation

Based on chapter 9

Relevance feedback

Relevance feedback is using **explicit or implicit user's input to** improve search results. Idea is to use this input as a **navigator in vector space** to drift towards better results.

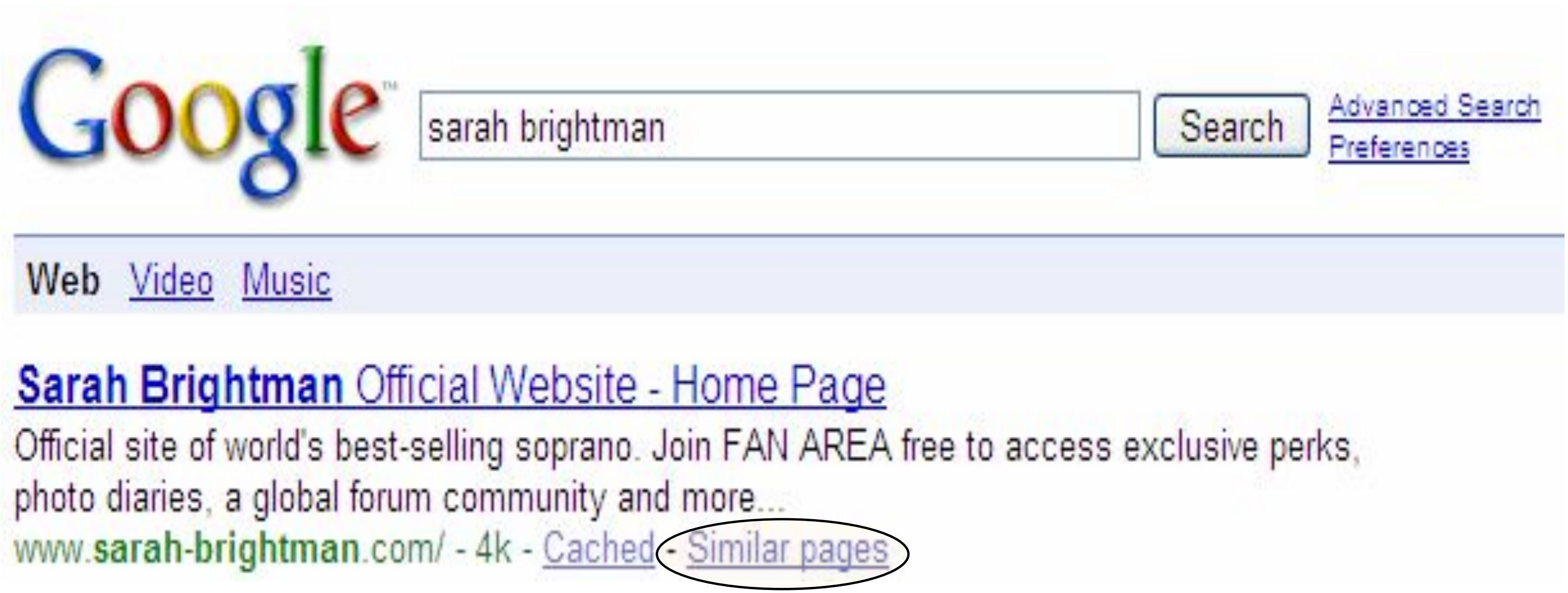
Relevance feedback

User **feedback** on relevance of docs in initial set of results:

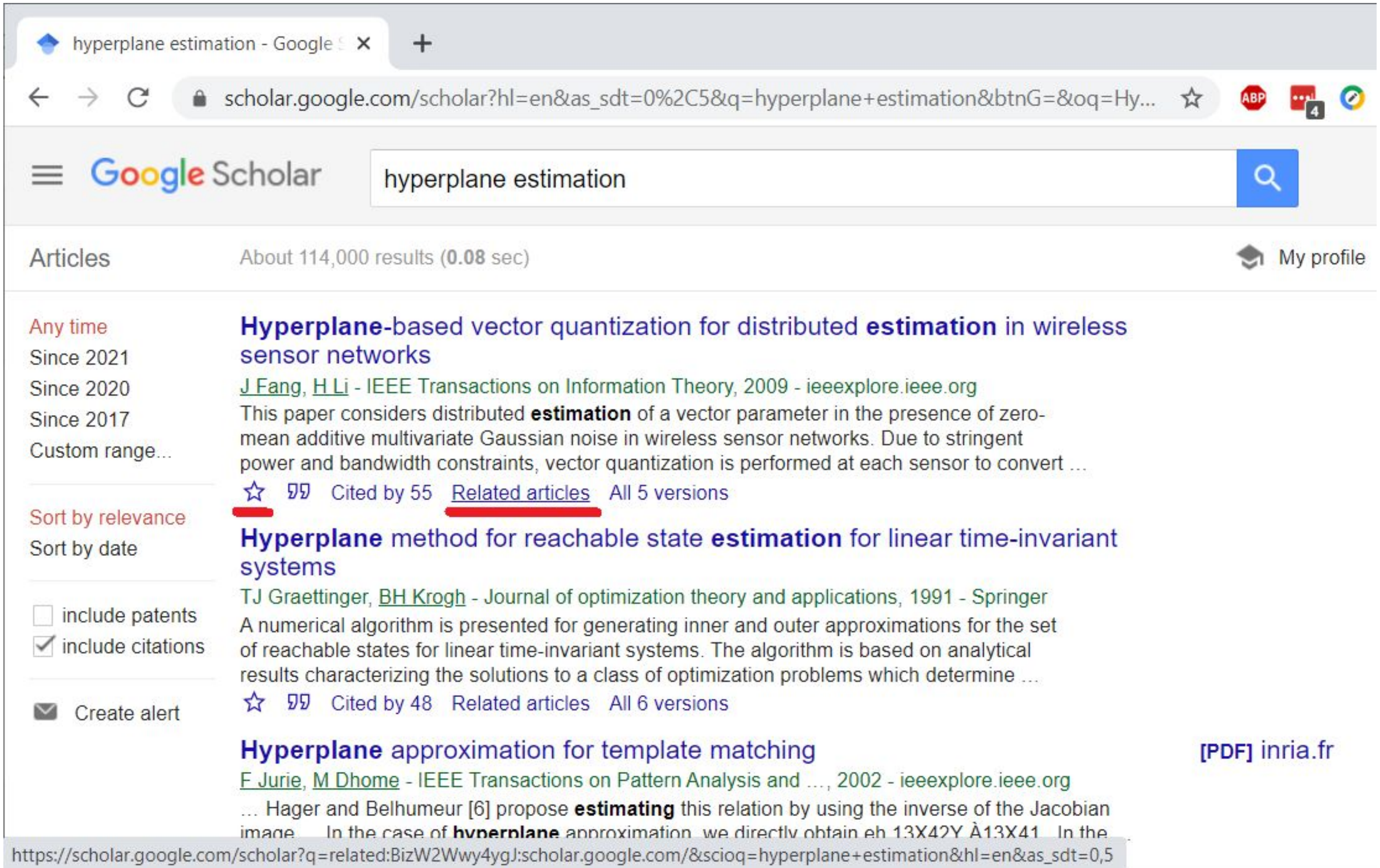
1. User issues a (short, simple) query
2. The user **marks** some results as relevant or non-relevant.
3. The IR system **computes** a better representation of the information need based on feedback.
4. Relevance feedback can go through one or more iterations.

Idea: it may be difficult to formulate a good query when you don't know the collection well, so iterate

Similar pages IRL 2009

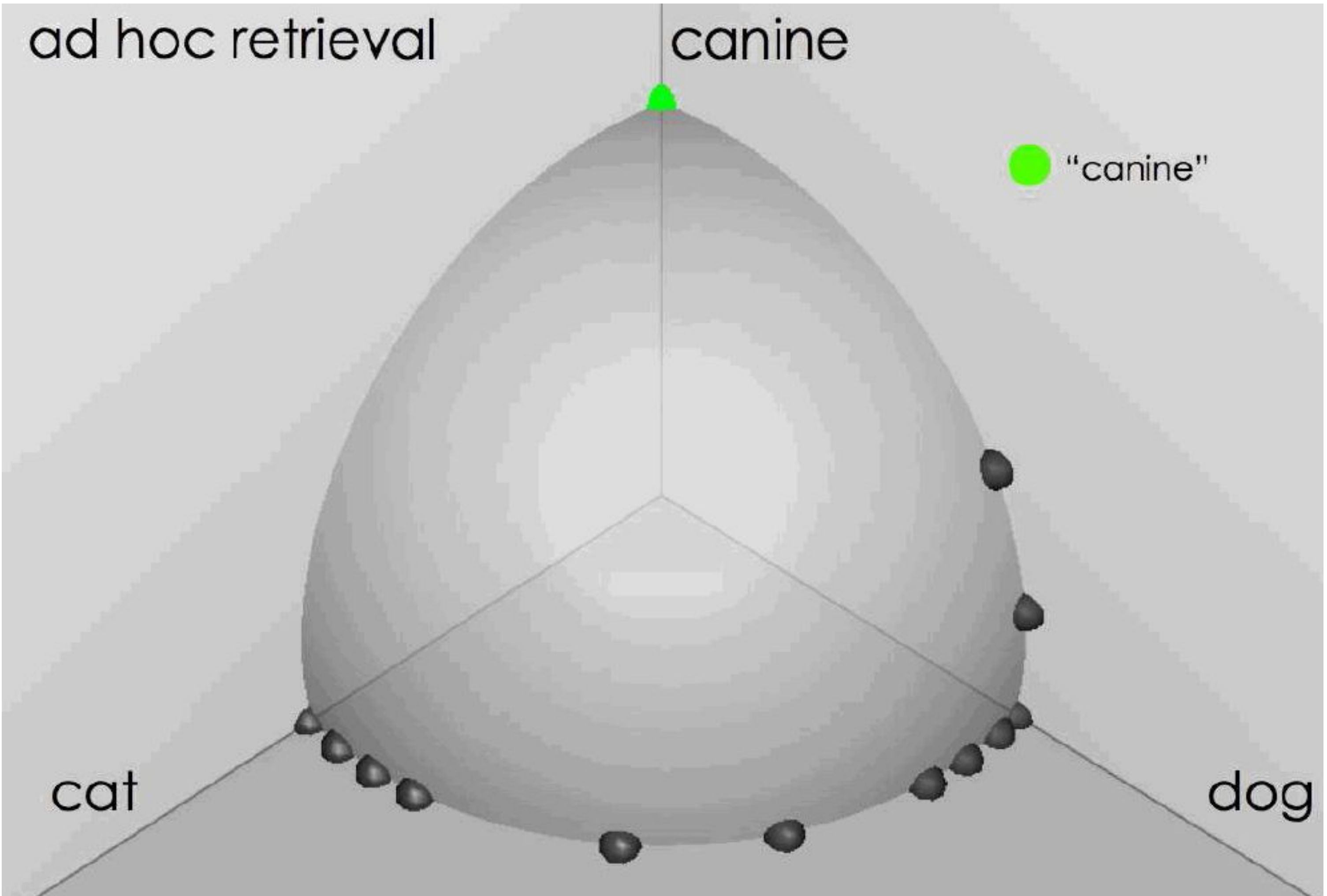


Similar pages IRL 2021



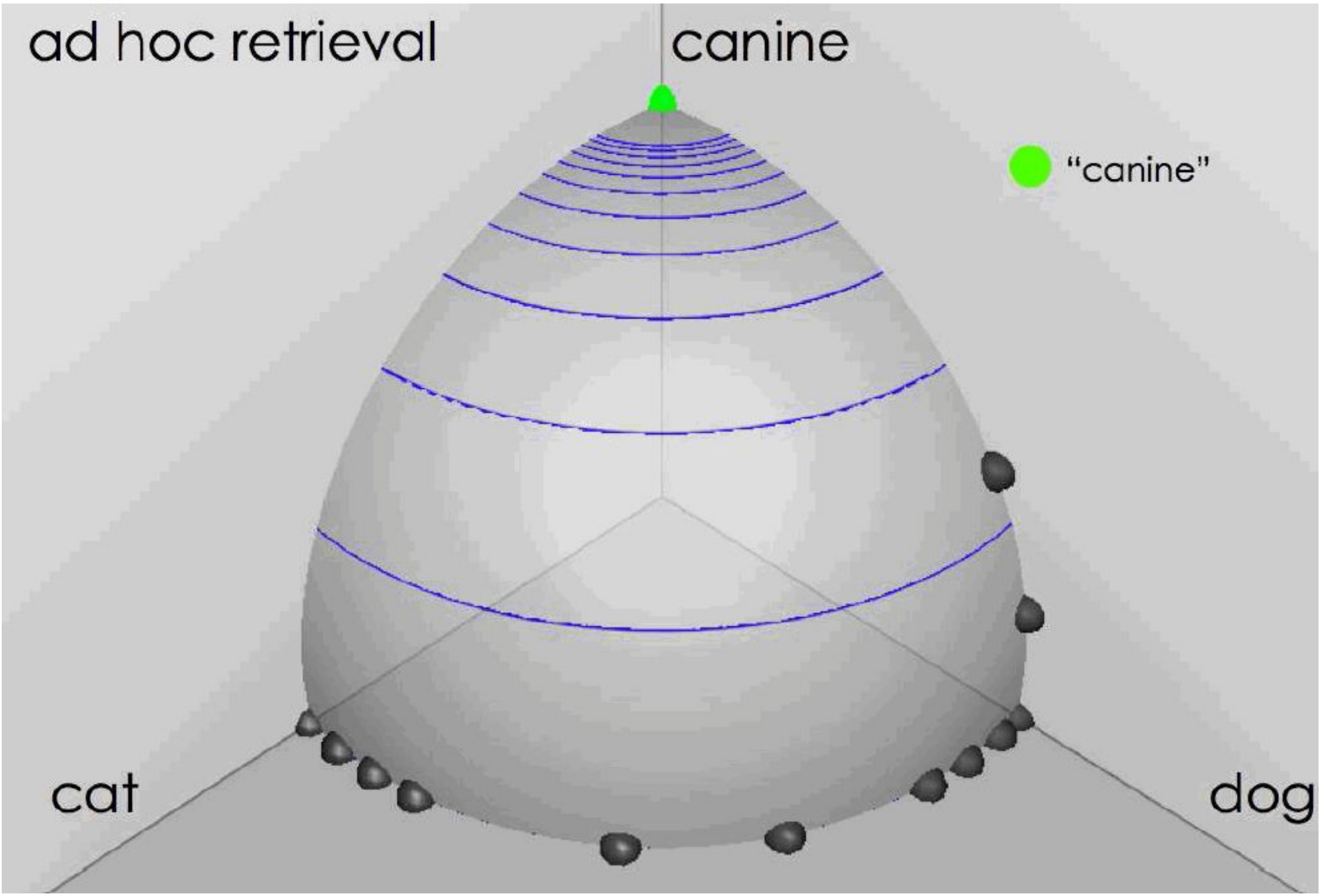
Ad hoc results for query *canine*

source: Fernando Diaz



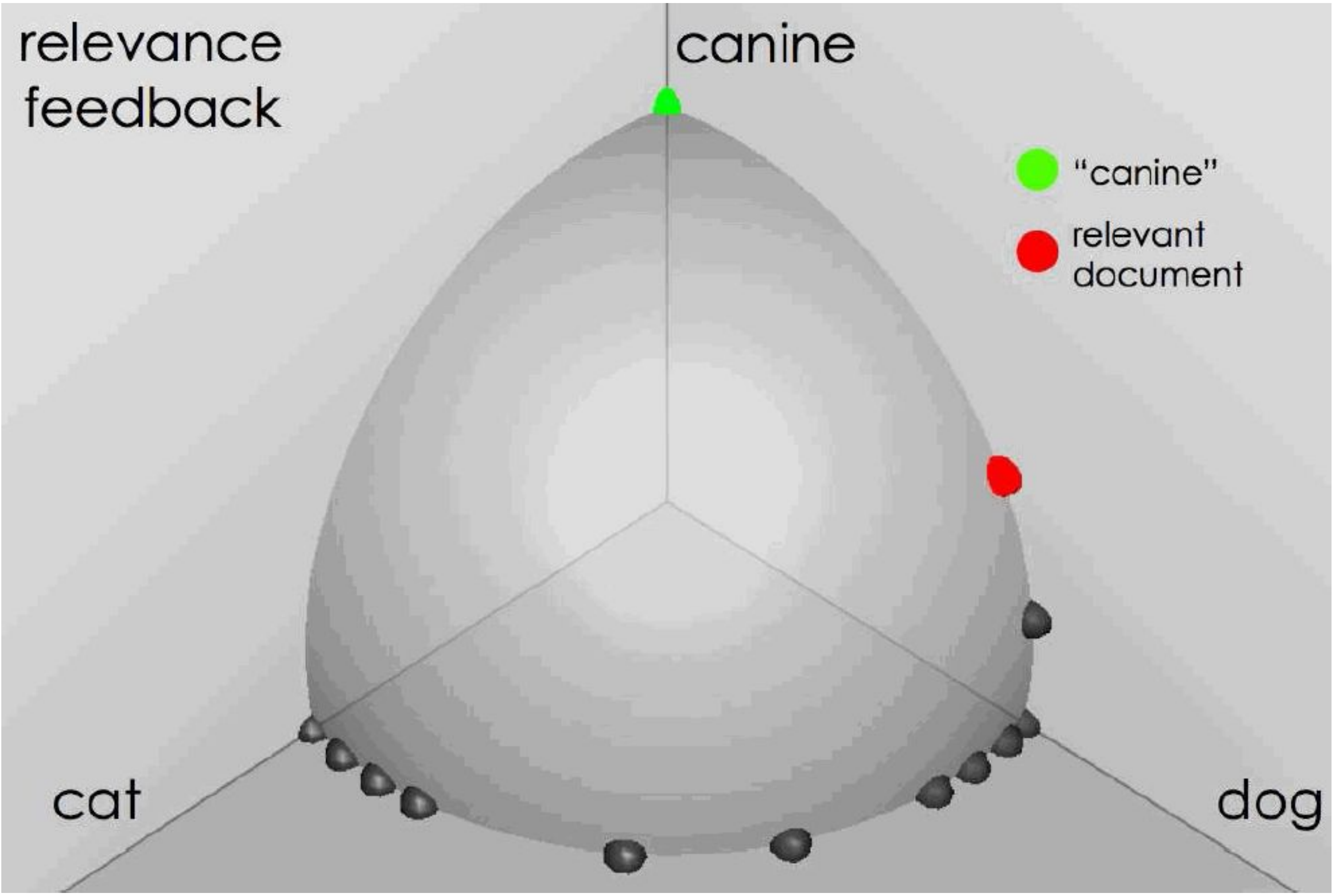
Ad hoc results for query *canine*

source: Fernando Diaz



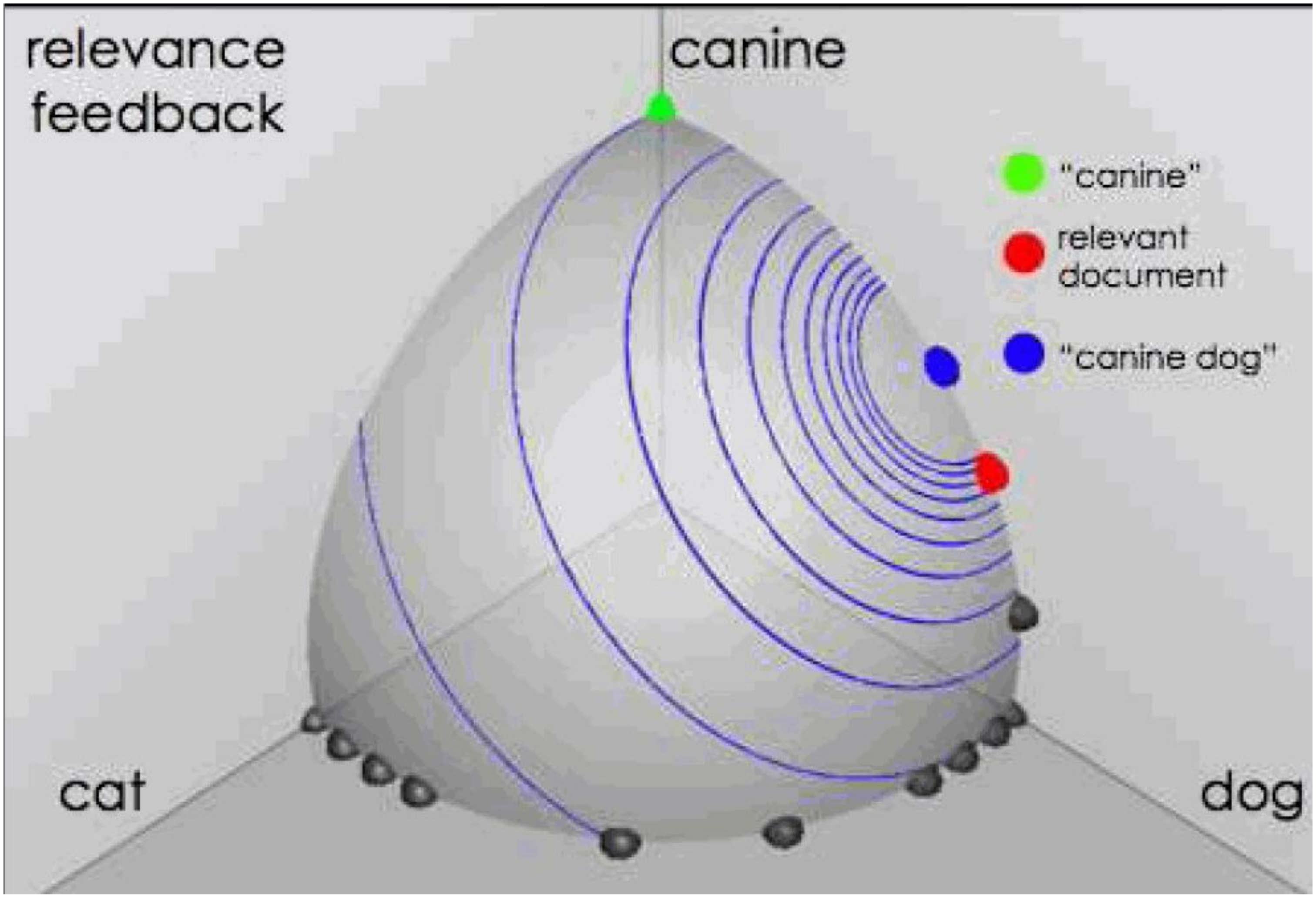
User feedback: Select what is relevant

source: Fernando Diaz



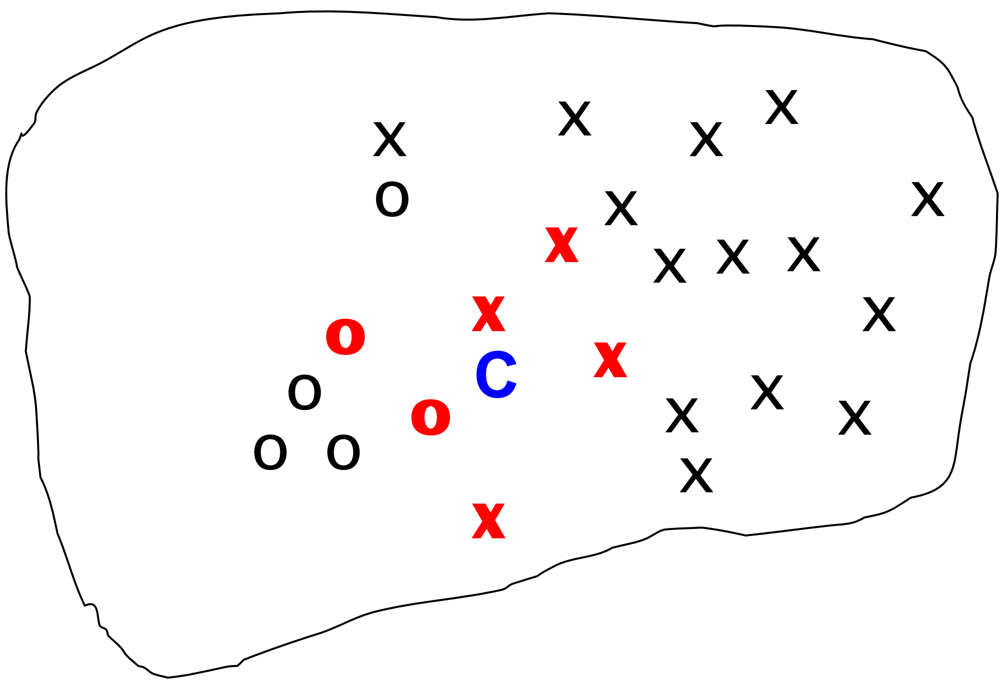
Results after relevance feedback

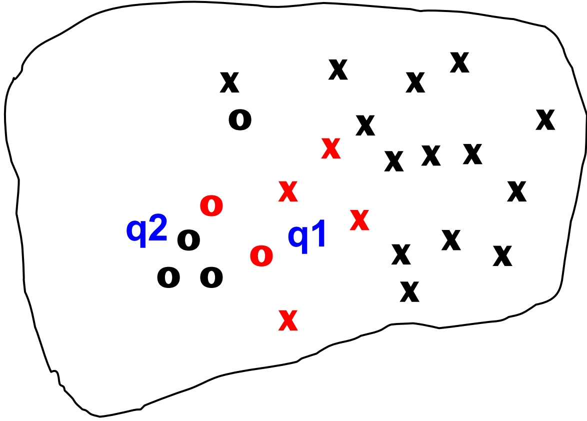
source: Fernando Diaz



Key concept: Centroid

- The <u>centroid</u> is the center of mass of a set of points (average vector)





Relevance feedback idea

- Uses the **vector space model** to pick a relevance feedback query
- Idea: move towards **relevant** and away from **non-relevant**
- Seek the query q_{opt} that maximizes

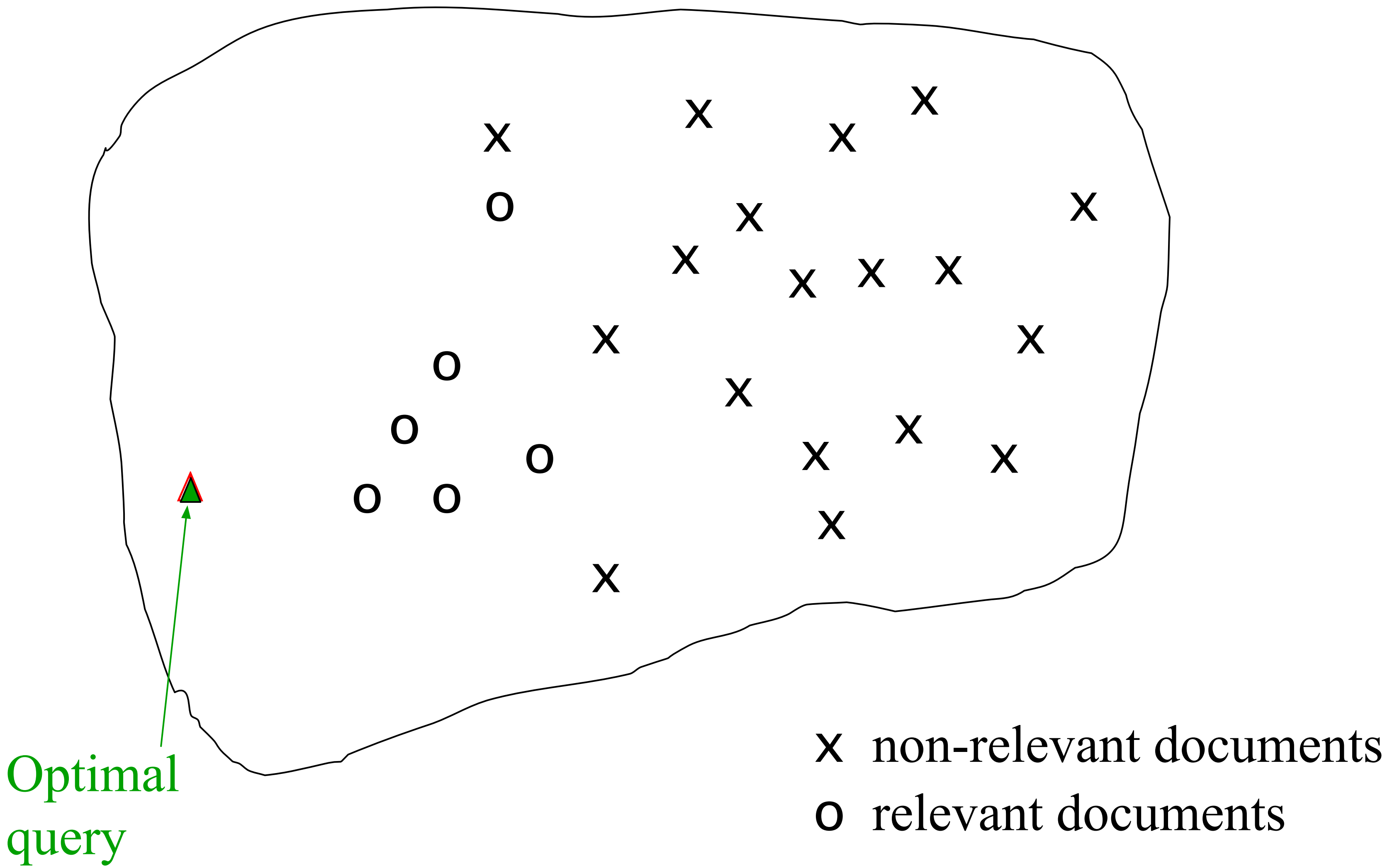
$$\vec{q}_{opt} = \arg\max_{\vec{q}} [\mathrm{sim}(\vec{q}, C_r) - \mathrm{sim}(\vec{q}, C_{nr})]$$

How?

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{|C_{nr}|} \sum_{\vec{d}_j \in C_{nr}} \vec{d}_j$$

Here C should be understood as a set of vectors described by a centroid (D later in book)

The Theoretically Best Query



Problems

1. We don’t know **all relevant** documents
2. We excluded **original query** out of consideration (q)
3. Will it bring us closer to relevant (*average relevant*), or we will jump over and leave a desired cluster (*average irrelevant*)?

Rocchio explicit algorithm

Kind of **regularization for relevance feedback**, which preserves from running away from relevant subspace and original query. Has **recommended parameter values**.

Rocchio 1971 Algorithm as a framework

- Used in practice:

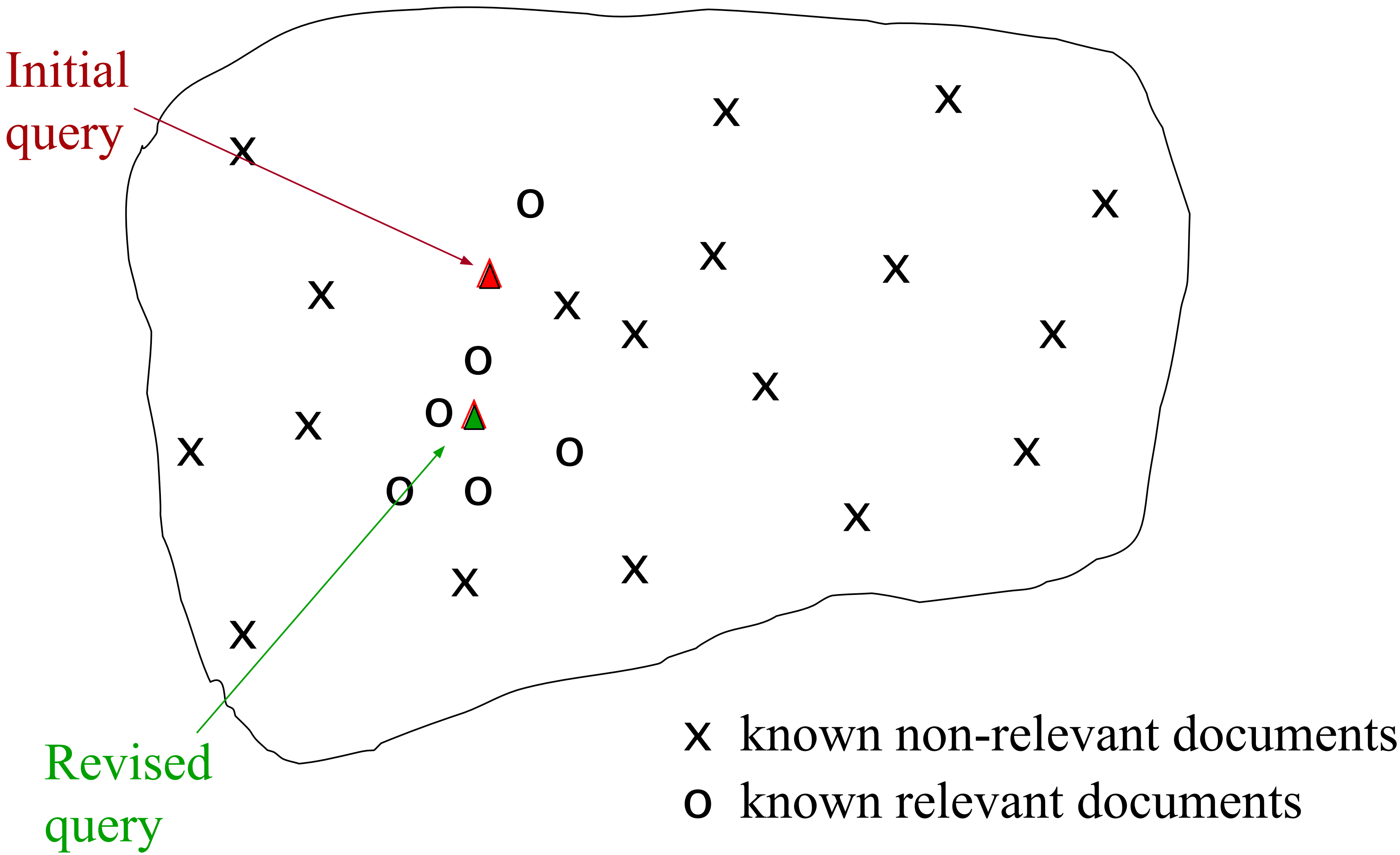
$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j\in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j\in D_{nr}}\vec{d}_j$$

- D_r = set of known relevant doc vectors
- D_{nr} = set of known irrelevant doc vectors
  - Different from C_r and C_{nr}
- q_m = modified query vector; q_0 = original query vector; α,β, γ: weights (hand-chosen or set empirically to **1, .75, .15**)
- New query **moves** toward relevant documents and away from irrelevant documents

Practical comments to framework

- Tradeoff α vs. β/γ : If we have a lot of judged documents, we want a higher β/γ.
- We can consider single most similar irrelevant document
- Mostly in practice improves recall, not precision

Relevance feedback on initial query



Relevance feedback overview

- We can modify the query based on relevance feedback and apply vector space model.
- Use only the docs that were marked.
- Relevance feedback can improve recall and precision
- Relevance feedback is most useful for increasing *recall* in situations where recall is important
  - Users can be expected to review results and to take time to iterate

Relevance feedback assumptions

- A1: User has **sufficient knowledge** for initial query.
- A2: Relevance prototypes are “well-behaved”.
  - Term distribution in relevant documents will be similar
  - Term distribution in non-relevant documents will be different from those in relevant documents
    - Either: All relevant documents are tightly clustered around a single prototype.
    - Or: There are different prototypes, but they have significant vocabulary overlap.
    - Similarities between relevant and irrelevant documents are small

Violation of A1

- User does not have sufficient initial knowledge.
- Examples:
  - Misspellings (Brittany Speers).
  - Cross-language information retrieval (гиперповерхность).
  - Mismatch of searcher’s vocabulary vs. collection vocabulary
    - Cosmonaut/astronaut

Violation of A2

- There are several relevance prototypes.
- Example:
  - **Pop stars** that worked at **Burger King**
  - **Kiev / Kyiv**
  - Different vocabularies

Relevance feedback problems

- Long queries are inefficient for typical IR engine.
- Users are often lazy to provide explicit feedback
- It’s often harder to understand why a particular document was retrieved after applying relevance feedback

Evaluation of relevance feedback

- Use q_0 and compute precision-recall graph
- Use q_m and compute precision-recall graph
  - Assess on all documents in the collection
    - **Spectacular improvements, but … it's cheating!**
    - Partly due to **known relevant documents** ranked higher
    - Must evaluate with respect to documents not seen by user
  - Use documents in residual collection (set of documents minus those assessed relevant)
    - Measures usually then **lower than for original query**
    - But a more realistic evaluation
    - Relative performance can be validly compared

Evaluation of relevance feedback

- Most satisfactory – use two collections each with their own relevance assessments
  - q_0 and user feedback from first collection
  - q_m run on second collection and measured
- **Empirically**, one round of relevance feedback is often very useful. Two rounds is sometimes marginally useful.

Pseudo and implicit feedbacks

User is lazy. Use **top search results** or **user search history** instead of explicit input to improve a query.

Pseudo relevance feedback

- Pseudo-relevance feedback automates the "**manual**" part of true relevance feedback.
- Pseudo-relevance 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.
- Several iterations will cause query drift.

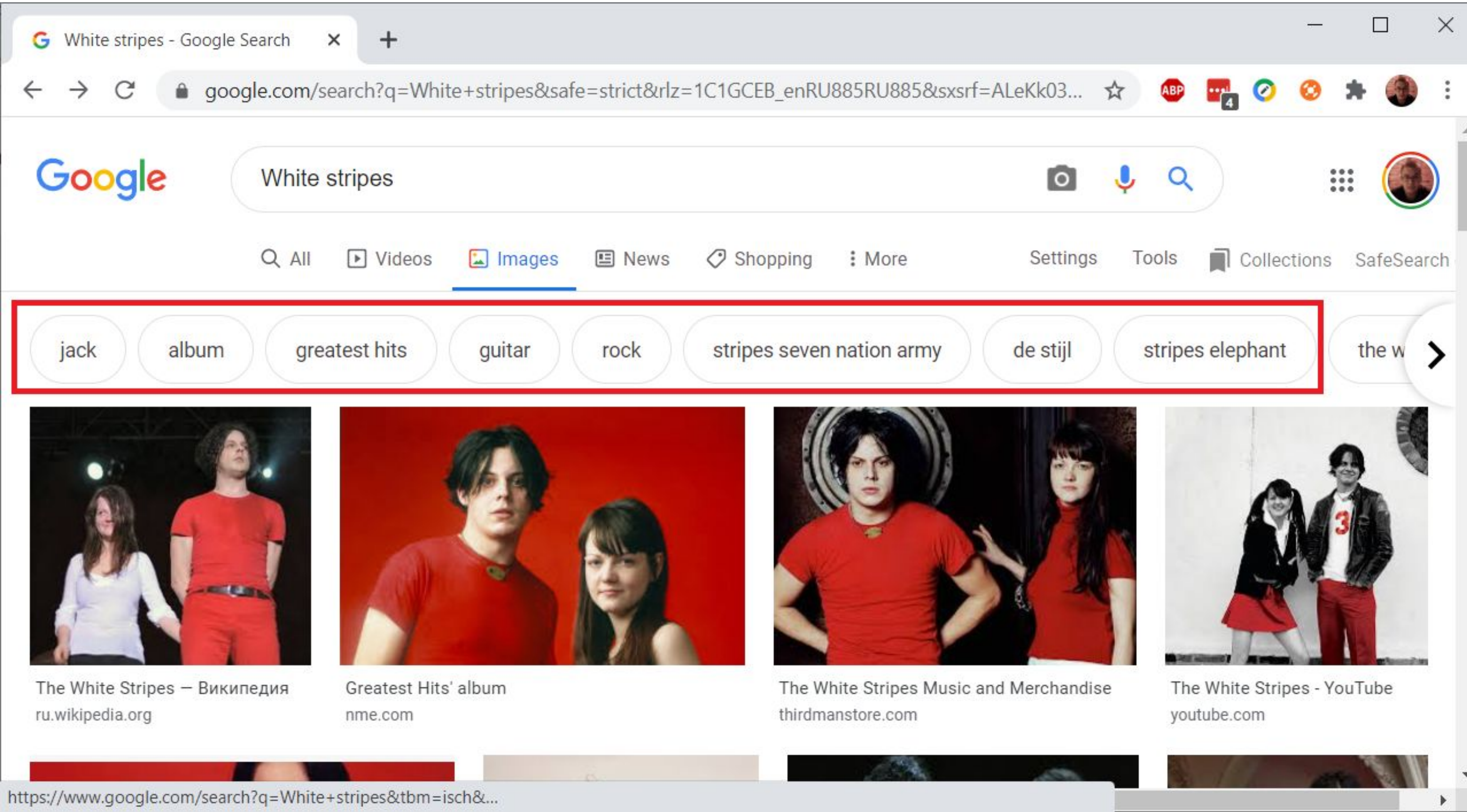
Implicit (indirect) relevance feedback

- Ok, we don't know the actual feedback
- But we know which documents user or users clicked for other queries
  - For a **single user** consider his/her preferences via CTR of the documents through other queries (e.g. "*How to trim a string*" for C++ developer)
  - For overall community select "*relevant*" based on high CTR

Query expansion and suggest

Query Expansion

- In relevance feedback, users give additional input (relevant/non-relevant) on **documents**, which is used to reweight terms in the documents
- In query expansion, users give additional input (good/bad search term) on **words or phrases**


White stripes - Google Search
google.com/search?q=White+stripes&safe=strict&rlz=1C1GCEB_enRU885RU885&sxsrf=ALeKk03...
Google
White stripes
All
Videos
Images
News
Shopping
More
Settings
Tools
Collections
SafeSearch
jack
album
greatest hits
guitar
rock
stripes seven nation army
de stijl
stripes elephant
The White Stripes — Википедия
ru.wikipedia.org
Greatest Hits' album
nme.com
The White Stripes Music and Merchandise
thirdmanstore.com
The White Stripes - YouTube
youtube.com
https://www.google.com/search?q=White+stripes&tbm=isch&...

How do we augment the user query?

- **Manual** thesaurus
  - E.g. MedLine: physician, syn: doc, doctor, MD, medico
  - Can be query rather than just synonyms
- Global Analysis: (static; of all documents in collection)
  - **Automatically derived thesaurus**
    - (co-occurrence statistics)
  - Refinements based on **query log** mining
    - Common on the web
- Local Analysis: (dynamic)
  - Analysis of documents in result set

Thesaurus-based auto query expansion

- For each term *t* in a query, expand the query with **synonyms** and **related words** of *t* from the thesaurus, maybe weighted
  - feline → feline +cat
- Generally increases *recall*
- Widely used in many science/engineering fields
- May significantly *decrease precision*, particularly with ambiguous terms.
  - "**interest** *rate*" → "interest rate +**fascinate** +*evaluate*"
- There is a high cost of manually producing a thesaurus
  - And for updating it for scientific changes

Automatic Thesaurus Generation

- Attempt to generate a thesaurus automatically by analyzing the collection of documents
- Fundamental notion: similarity between two words
- *Definition 1: Two words are similar if they co-occur with similar words.*
- *Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.*
  - **Co-occurrence** based is more **robust**,
  - **grammatical relations** are more **accurate**.

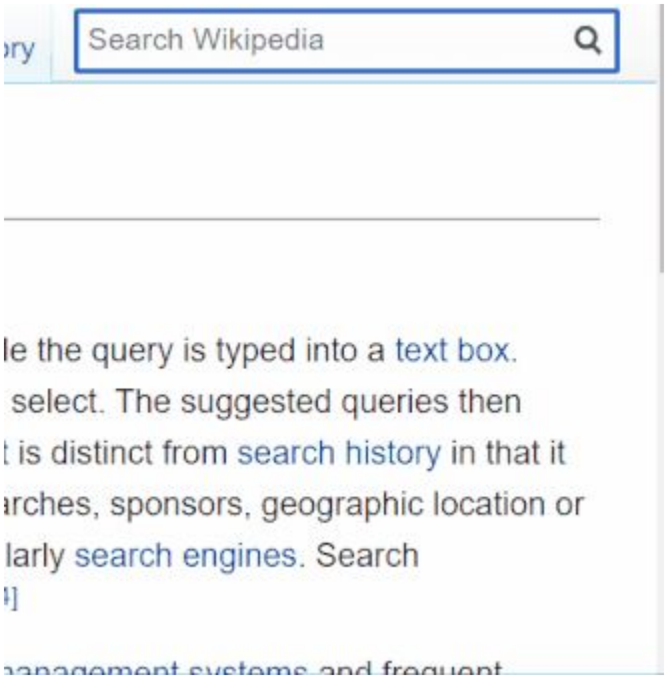
Automatic Thesaurus Generation Discussion

- **Quality of associations** is usually a problem.
- Term **ambiguity** may introduce irrelevant statistically correlated terms.
    - "Apple computer" → "Apple +red +fruit computer"
- Since terms are **highly correlated** anyway, expansion may **not retrieve** many **additional** documents.

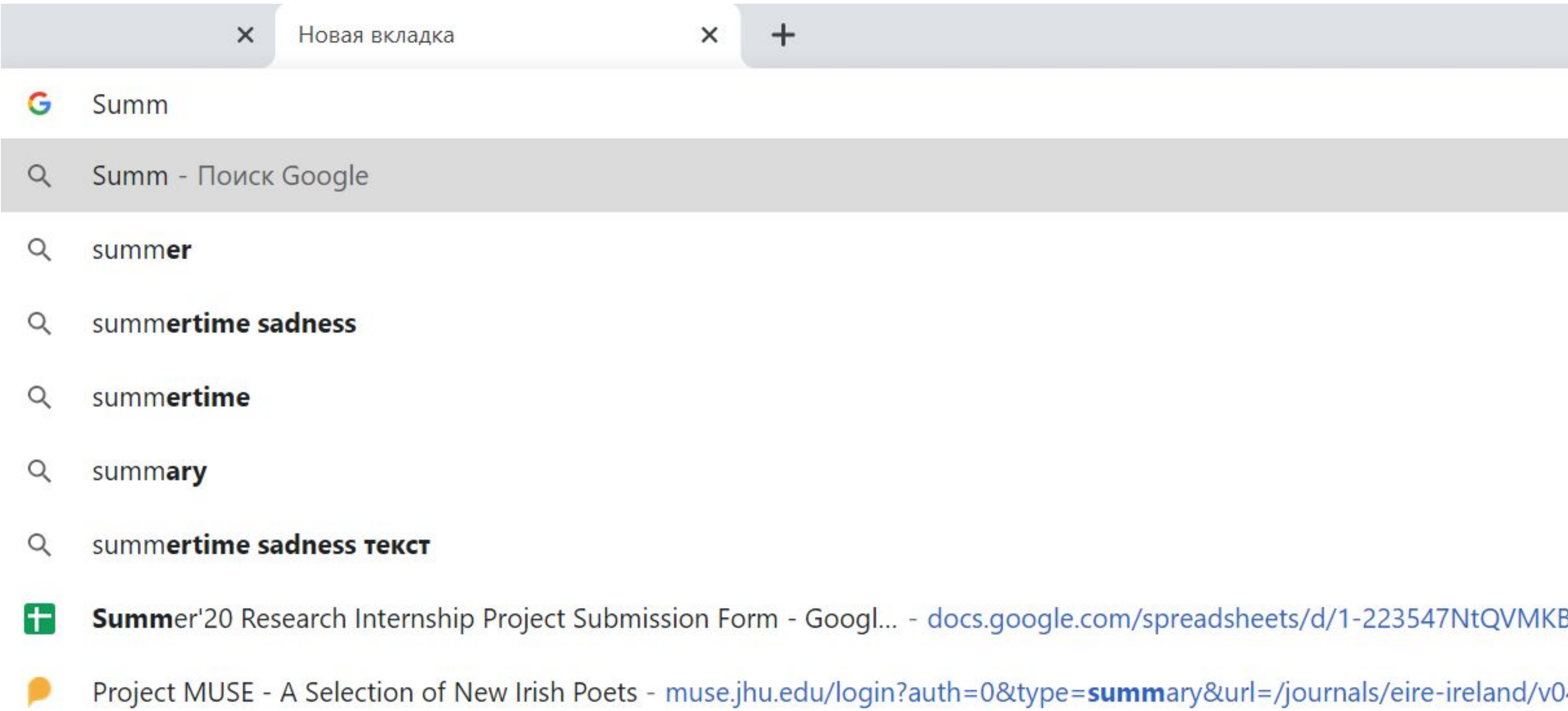
Suggest

... query feature used in computing to show the **searcher shortcuts**, while the query is typed into a text box. Before the query is complete, a drop-down list with the **suggested completions** appears to provide options to select [wiki]

- **Blacklist** of what can be a “bad” suggest
- **Complaints** on certain suggestions (bots, law violations, insults)
- **Trie** is the most useful data structure to Implement suggestions



Suggest



Новая вкладка
Summ
Summ - Поиск Google
summer
summertime sadness
summertime
summary
summertime sadness текст
Summer'20 Research Internship Project Submission Form - Googl... - docs.google.com/spreadsheets/d/1-223547NtQVMKB
Project MUSE - A Selection of New Irish Poets - muse.jhu.edu/login?auth=0&type=summary&url=/journals/eire-ireland/v0

Thanks for your attention!