

# Расширение запроса при поиске

Маннинг и др. Введение в  
информационный поиск, гл.9

# Методы расширения запроса

- Несовпадение слова запроса:
  - самолет – лайнер
- Методы расширения запроса:
  - Глобальные методы
    - Ручные тезаурусы
    - Автоматически порождаемый тезаурус
  - Локальные методы (по конкретному запросу)
    - Relevance feedback (обратная связь по релевантности)
    - Pseudo Relevance feedback (обратная связь по псевдорелевантности)

# Тезаурус

- Разные виды
- Словарь с формализованными связями между словами

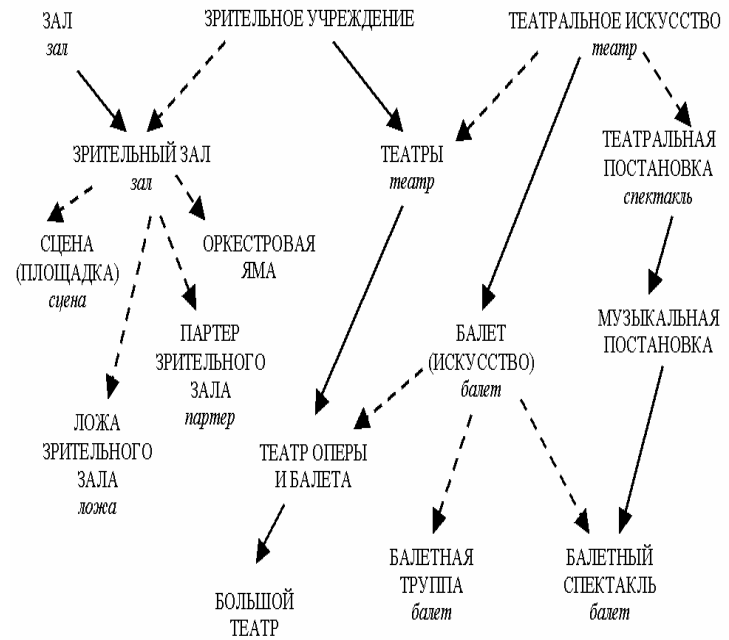
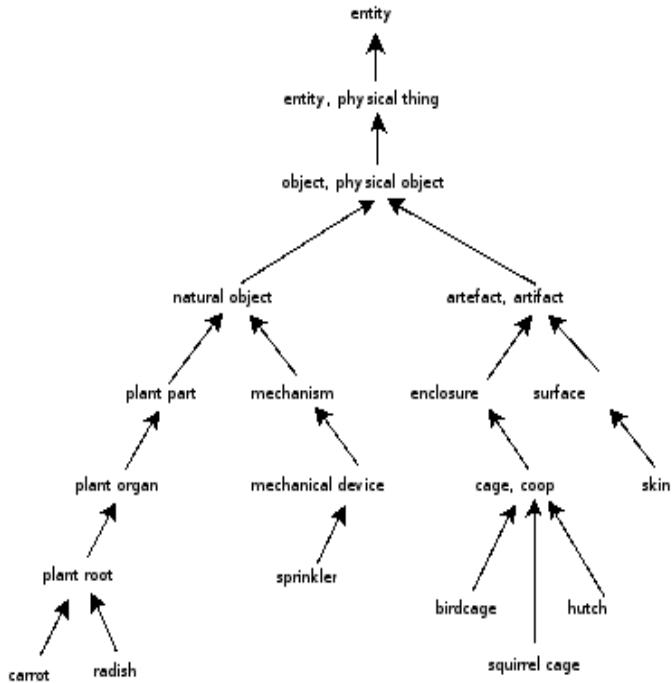


Figure 1. "is a" relation example

# Исторические этапы в расширении запросов













- Булевский поиск: короткие тексты (1965)
  - Информационно-поисковые тезаурусы, ручное индексирование
- Векторные модели (1991-1992)
  - Локальное расширение запросов: relevance feedback
- Публикация тезауруса WordNet (1995)
  - Комбинирование с векторными моделями
- Автоматически порождаемые тезаурусы (2010)
  - Google, Yandex
- Рост значимости поведенческих факторов (2015)
- Использование тезаурусов в предметно-ориентированных системах

# Обратная связь по релевантности

- Пользователь оценивает документы в поисковой выдаче
  - Пользователь задает относительно простой, короткий запрос
  - Затем пользователь размечает часть результатов как релевантные и нерелевантные
  - Система вычисляет улучшает соответствие документов запросу на основе пользовательской разметки
  - Процедура может выполняться итеративно.
- Основная идея: сформулировать хороший запрос трудно, если пользователь не знаком с коллекцией, поэтому – итеративное построение запроса


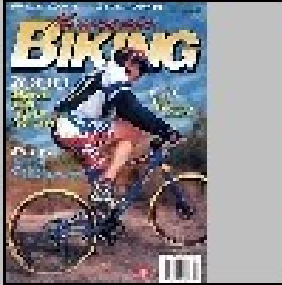



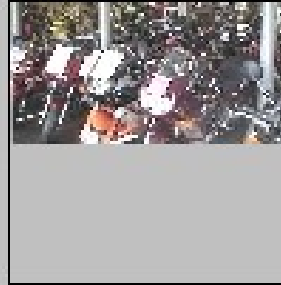






# Результаты для начального запроса

[Browse](#) [Search](#) [Prev](#) [Next](#) [Random](#)

|  |   |   |  |   |   |
|--|---|---|--|---|---|
| <br>(144473, 16458)<br>0.0<br>0.0<br>0.0   | <br>(144457, 252140)<br>0.0<br>0.0<br>0.0  | <br>(144456, 262857)<br>0.0<br>0.0<br>0.0  | <br>(144456, 262863)<br>0.0<br>0.0<br>0.0  | <br>(144457, 252134)<br>0.0<br>0.0<br>0.0  | <br>(144483, 265154)<br>0.0<br>0.0<br>0.0  |
| <br>(144483, 264644)<br>0.0<br>0.0<br>0.0 | <br>(144483, 265153)<br>0.0<br>0.0<br>0.0 | <br>(144518, 257752)<br>0.0<br>0.0<br>0.0 | <br>(144538, 525937)<br>0.0<br>0.0<br>0.0 | <br>(144456, 249611)<br>0.0<br>0.0<br>0.0 | <br>(144456, 250064)<br>0.0<br>0.0<br>0.0 |

# Разметка пользователя

[Browse](#) [Search](#) [Prev](#) [Next](#) [Random](#)

|   |  |  |   |  |  |
|---|--|--|---|--|--|
|   |   |   |   |   |   |
| (144473, 16458)<br>0.0<br>0.0<br>0.0  | (144457, 252140)<br>0.0<br>0.0<br>0.0  | (144456, 262857)<br>0.0<br>0.0<br>0.0  | (144456, 262863)<br>0.0<br>0.0<br>0.0   | (144457, 252134)<br>0.0<br>0.0<br>0.0  | (144483, 265154)<br>0.0<br>0.0<br>0.0  |
|  |  |  |  |  |  |
| (144483, 264644)<br>0.0<br>0.0<br>0.0   | (144483, 265153)<br>0.0<br>0.0<br>0.0  | (144518, 257752)<br>0.0<br>0.0<br>0.0  | (144538, 525937)<br>0.0<br>0.0<br>0.0   | (144456, 249611)<br>0.0<br>0.0<br>0.0  | (144456, 250064)<br>0.0<br>0.0<br>0.0  |

# Результаты после разметки

[Browse](#)
[Search](#)
[Prev](#)
[Next](#)
[Random](#)


(144538, 523493)  
0.54182  
0.231944  
0.309876



(144538, 523835)  
0.56319296  
0.267304  
0.295889



(144538, 523529)  
0.584279  
0.280881  
0.303398



(144456, 253569)  
0.64501  
0.351395  
0.293615



(144456, 253568)  
0.650275  
0.411745  
0.23853



(144538, 523799)  
0.66709197  
0.358033  
0.309059



(144473, 16249)  
0.6721  
0.393922  
0.278178



(144456, 249634)  
0.675018  
0.4639  
0.211118



(144456, 253693)  
0.676901  
0.47645  
0.200451



(144473, 16328)  
0.700339  
0.309002  
0.391337



(144483, 265264)  
0.70170796  
0.36176  
0.339948

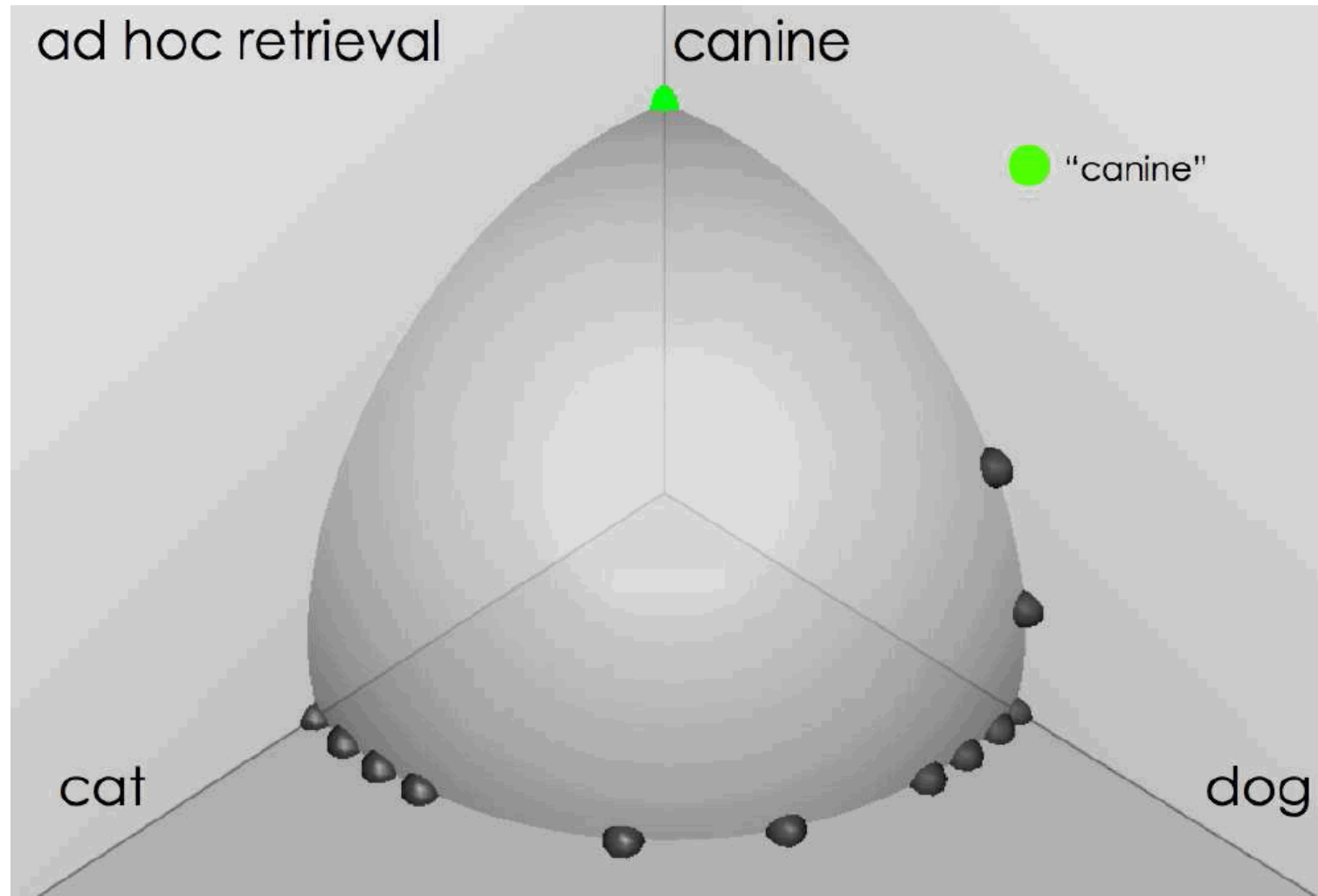


(144478, 512410)  
0.70297  
0.469111  
0.233859



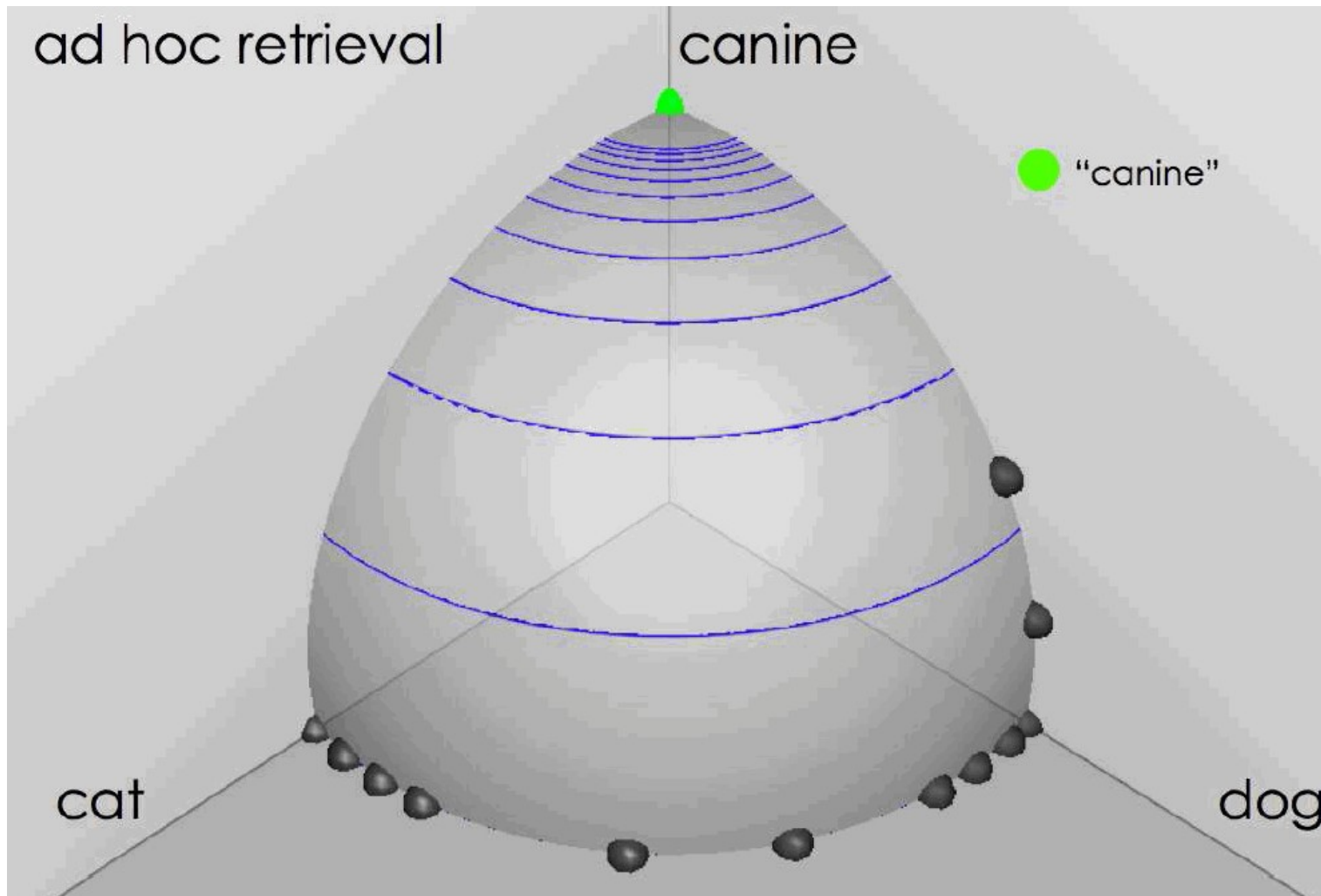
# Выдача по запросу *canine*

source: Fernando Diaz



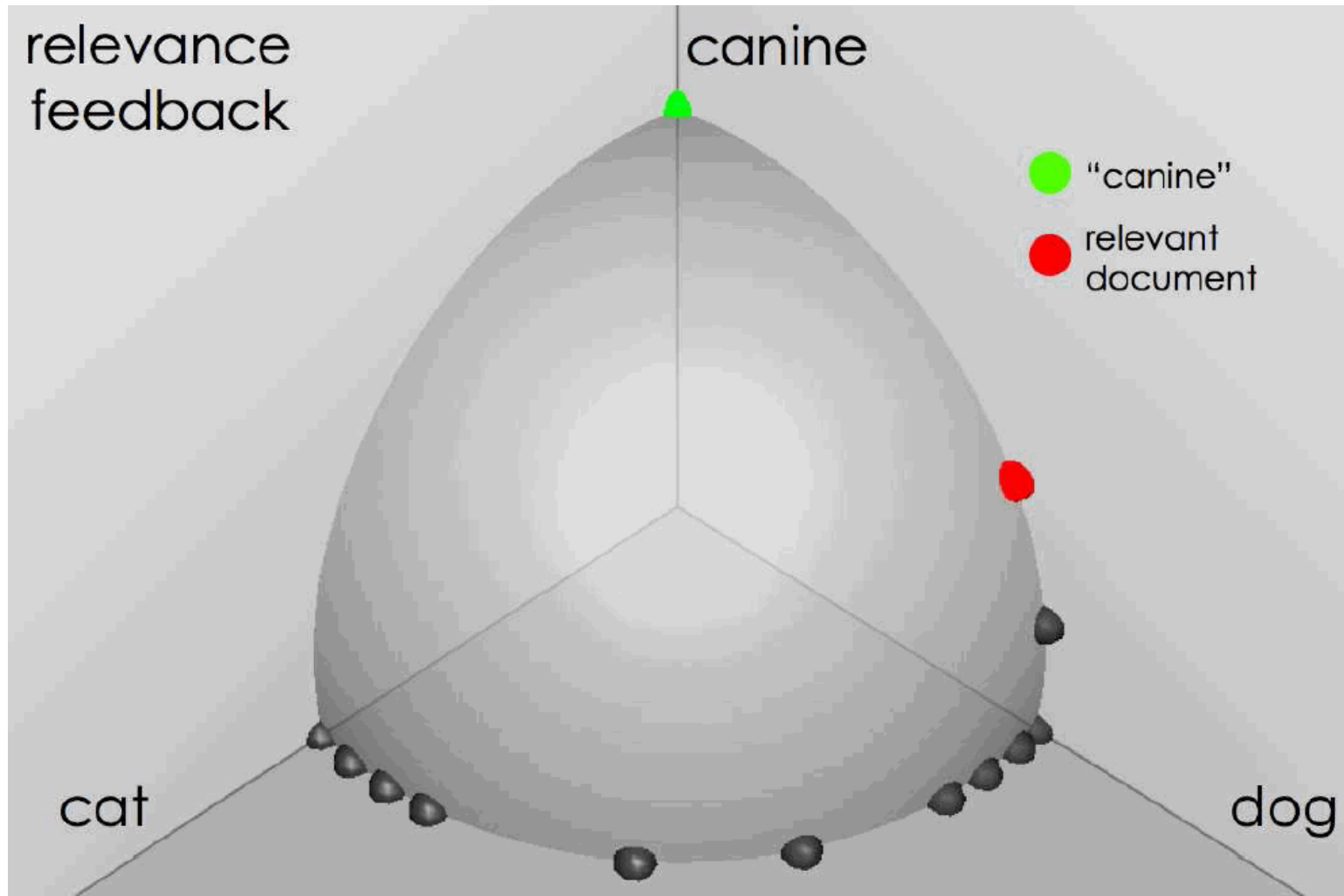
# Выдача по запросу *canine-2*

source: Fernando Diaz



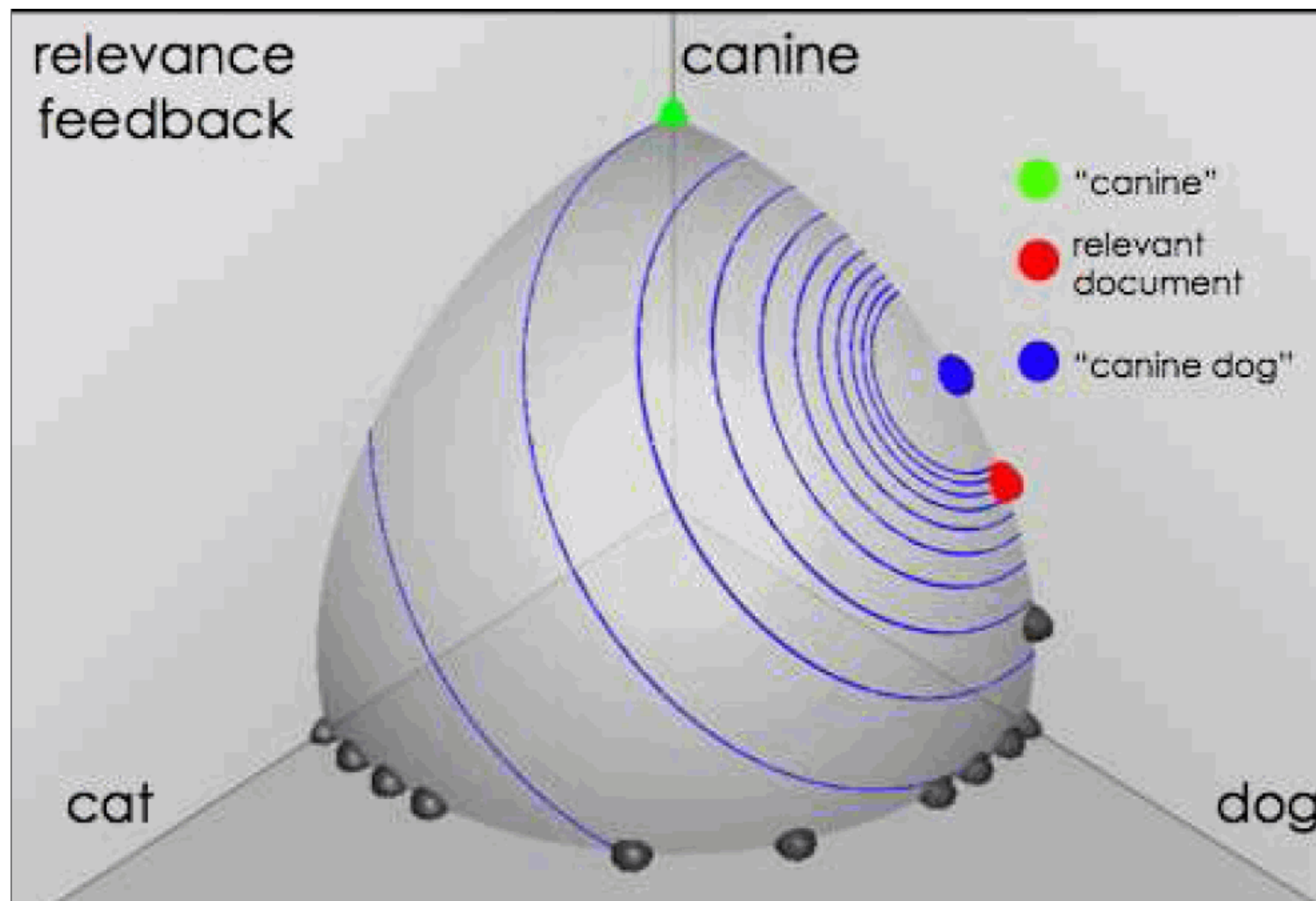
# Пользователь выбирает релевантное

source: Fernando Diaz



# Результаты (relevance feedback)

source: Fernando Diaz



# Начальный запрос и результаты

- Запрос: *New space satellite applications*
  1. 0.539, 08/13/91, [NASA Hasn't Scrapped Imaging Spectrometer](#)
  2. 0.533, 07/09/91, [NASA Scratches Environment Gear From Satellite Plan](#)
  3. 0.528, 04/04/90, [Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes](#)
  4. 0.526, 09/09/91, [A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget](#)
  5. 0.525, 07/24/90, [Scientist Who Exposed Global Warming Proposes Satellites for Climate Research](#)
  6. 0.524, 08/22/90, [Report Provides Support for the Critics Of Using Big Satellites to Study Climate](#)
  7. 0.516, 04/13/87, [Arianespace Receives Satellite Launch Pact From Telesat Canada](#)
  8. 0.509, 12/02/87, [Telecommunications Tale of Two Companies](#)
- Пользователь отмечает релевантные результаты отметкой “+”.

## Расширенные запрос после relevance feedback

- 2.074 new      15.106 space
- 30.816 satellite      5.660 application
- 5.991 nasa      5.196 eos
- 4.196 launch      3.972 aster
- 3.516 instrument      3.446 arianespace
- 3.004 bundespost      2.806 ss
- 2.790 rocket      2.053 scientist
- 2.003 broadcast      1.172 earth
- 0.836 oil      0.646 measure

## Ключевое понятие: центроид

- Центроид – это центр масс совокупности точек
- Документы – это точки в многомерном пространстве
- Определение: Центроид

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

где  $C$  – множество документов.

# Алгоритм Роккьо (Rocchio)

- Алгоритм Rocchio использует векторное пространства найти наилучший запрос на основе пользовательской разметки
- Rocchio ищет запрос  $q_{opt}$ , который максимизирует

$$q_{opt} = \arg \max_{\vec{q}} [\cos(q, \mu(C_r)) - \cos(q, \mu(C_{nr}))]$$

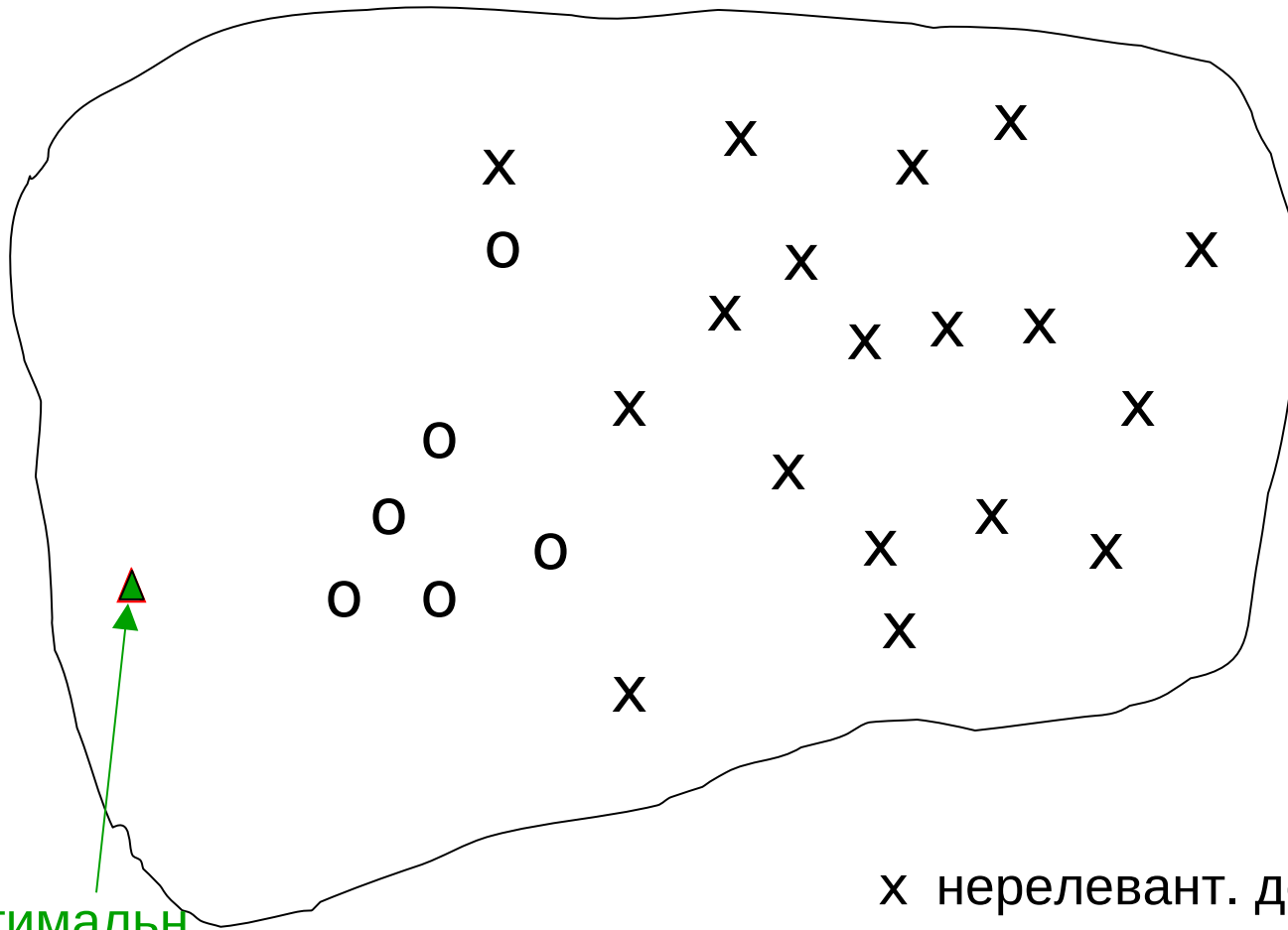
- Пытается отделить релевантные и нерелевантные документы

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

- Проблема: мы не знаем все релевантные документы



# Лучший запрос

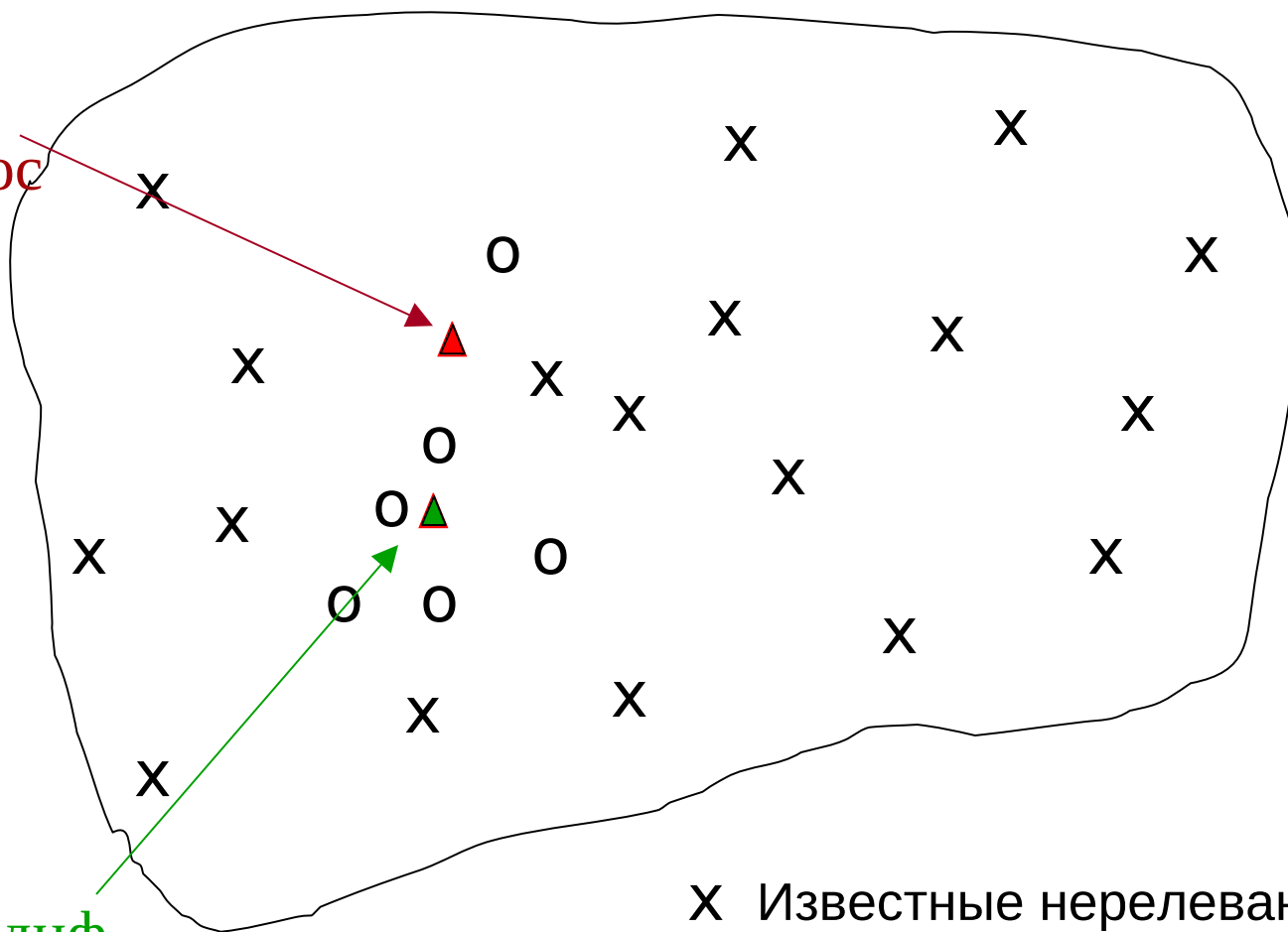


Оптимальн.  
запрос

x нерелевант. документы  
o релевантные документы

# Relevance feedback по исходному запросу

Исх.  
запрос



X Известные нерелевантн. док-ты  
o Известные релевантные док-ты

Модиф.  
запрос

# Rocchio 1971 алгоритм (SMART)

- На практике используется:

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

- $D_r$  = множество известных релевантных doc векторов
- $D_{nr}$  = множество известных нерелевантных doc векторов

– Отличны от  $C_r$  и  $C_{nr}$



- $q_m$  = модифицированный вектор запроса;  $q_0$  = исходный вектор запроса;  $\alpha, \beta, \gamma$ : веса
- Новый запрос «сдвигается» по направлению к релевантным документам и «уходит» от нерелевантных документов

## Особенности параметров

- Соотношение  $\alpha$  vs.  $\beta/\gamma$  : Если у нас много оцененных документов, то лучше более высокие  $\beta/\gamma$ .
- Некоторые веса в модифицированном векторе запроса становятся отрицательными
  - Отрицательные веса слов игнорируются (устанавливаются равными 0)

# Relevance Feedback

## в векторных пространствах

- Можно модифицировать запрос на основе разметки пользователя и применить стандартную векторную модель.
- Используются только документы, которые размечены.
- Relevance feedback может улучшить и полноту и точность
- Relevance feedback наиболее полезен в увеличении полноты в тех ситуациях, когда полнота важна
  - Пользователи должны просматривать и размечать результаты
    - Несколько итераций

# Позитивный vs Негативный Feedback

- Позитивный feedback более ценен, чем негативный feedback (обычно  $\gamma < \beta$ ; например,  $\gamma = 0.25$ ,  $\beta = 0.75$ ).
- Многие системы позволяют только позитивный feedback ( $\gamma=0$ ).

# Relevance Feedback: предположения

- A1: Пользователь имеет достаточно знаний для исходного запроса
- A2: Прототипы релевантных/нерелевантных документов “ведут себя хорошо”
  - Распределение слов в релевантных документах сходно
  - Распределение слов в нерелевантных документах отлично от распределения слов в релевантных документах
    - 1) Все релевантные документы похожи на один прототип
    - 2) Имеется несколько прототипов, но у них значительное пересечение по составу
    - Сходство между релевантными и нерелевантными документами относительно небольшое

# Нарушение A1

- У пользователя нет достаточного начального знания
- Примеры:
  - Неправильное написание: Brittany Speers.
  - Многоязыковой информационный поиск (hígado).
  - Несоответствие словаря пользователя и словаря коллекции
    - Cosmonaut/astronaut



## Нарушение A2

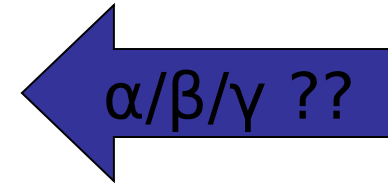
- Имеется несколько прототипов
  - Поэтому между текстами может быть мало общего
  - Например, нужно найти, где учились известные актеры
- Часто: примеры более общего понятия

# Relevance Feedback: Проблемы

- Длинные запросы – неэффективны для типичной поисковой машины
  - Большое ожидание для пользователя
  - Высокая стоимость для поисковой системы
  - Частичное решение:
    - Использование только слов с наиболее высоким весом
      - Например, 20 первых по весу
- Пользователи часто не хотят размечать документы
- Трудно понять, почему данный документ был выдан после relevance feedback

# Relevance Feedback в Интернет

- Некоторые поисковые машины предлагают возможность просмотра похожих страниц
  - Тривиальная форма relevance feedback
  - Google (link-based)
  - Altavista
  - Stanford WebBase
- Но результаты трудно объяснить среднему пользователю
- Excite
  - вводил настоящий relevance feedback,
  - затем убрал – никто не пользовался



# Pseudo relevance feedback

- Pseudo-relevance feedback автоматизирует «ручную» часть реального relevance feedback.
- Pseudo-relevance алгоритм:
  - Строит поисковую выдачу по запросу
  - Предполагает, что первые  $k$  документов - релевантны
  - Выполняет relevance feedback
- В среднем хорошо работает
- Но может получить очень плохие результаты для некоторых запросов
- Несколько итераций могут вызвать «искажение запроса»

# Задание 5: Задача 3

- Запрос: отбор кандидатов
- Пользователь отметил релевантными два документа
  - Кандидат отобрать претендент
  - Отбор выбрать претендент
- Объем коллекции – 1 млн. документов
- Df:
  - отбор 70000, кандидат – 70000,
  - Претендент - 30000, отобрать – 50000, выбрать 70000
- Как изменится запрос, если
  - $\alpha=0.7$  (коэффициент учета запроса),
  - $\beta=0.3$  (коэффициент учета релевантных документов),
  - Запрос представляется как вектор частот (count)
  - Документ представляется как нормализованный вектор tf.idf
    - $Tf=count$

# Тест

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{q_i}{|\vec{q}|} \cdot \frac{d_i}{|\vec{d}|} = \frac{\sum q_i d_i}{\sqrt{\sum q_i^2} \sqrt{\sum d_i^2}}$$

- Если при подсчете косинусной меры сходства запроса с документом забыть разделить на длину запроса, то
  - а. Вперед продвинутся документы, в которых запрос больше совпадает с заголовком
  - б. Порядок выдачи документов не изменится
  - с. Вперед продвинутся более короткие документы
  - д. Вперед продвинутся более длинные документы