LC029 정보검색

2022년 11월 10일

Chapter 9 : Relevance Feedback and Query Expansion

For High Recall

- The same concept may be represented with different words. 같은 개념을 지칭하는 서로 다르게 생긴 단어를 검색하면 관련된 결과를 잘 찾아주기 힘들다
  - Searching for *aircraft* doesn't match with *plane*
  - Searching for *thermodynamic* doesn't match with *heat*
- Solutions for High Recall 해결 방법
  - Manual refinement of a query 수동 질의 확장
  - Automatic refinement of a query 자동 질의 확장
- We will discuss the automatic refinement methods.

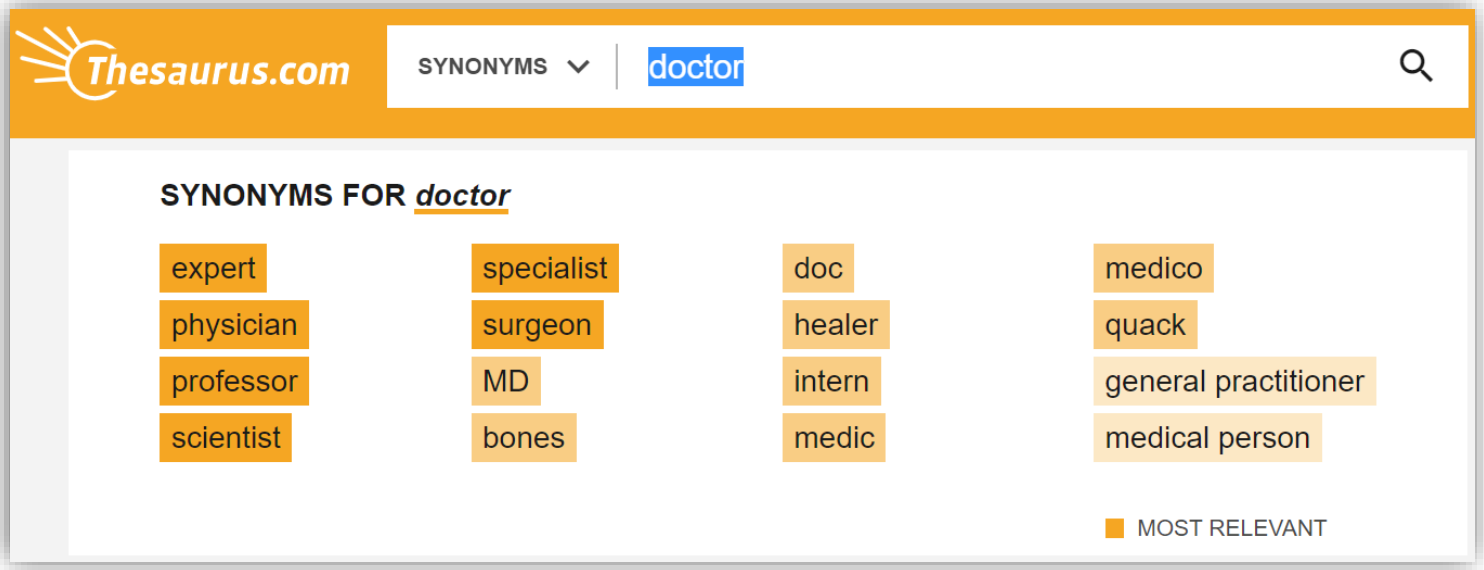
Automatic Refinement of Queries 자동 질의 확장

- Global methods (static) 비슷한 단어로 질의 확장
  - Expanding query terms with semantically similar terms. 유의어 사전이 필요함
  - Query expansion with a **thesaurus** or **WordNet**
  - Query expansion with **automatically generated thesaurus**
- Local methods (dynamic) 사용자가 검색 결과가 잘 나왔는지 피드백 후 재검색
  - Refining a query **relative to the documents** that initially appear to match the query.
  - **Relevance feedback**
  - **Pseudo relevance feedback**
  - **Indirect relevance feedback**

Thesaurus 유의어/반의어 사전 th ㅣ쏘ㄹ로스라고 읽으면 됨 더사우루스 아님

- From Wikipedia,
  - A thesaurus or synonym dictionary is a reference work for finding synonyms and sometimes antonyms of words.
  - Often used by writers to help find the best word to express an idea:

    … to find the word or words, by which an idea may be most fitly and aptly expressed (by Peter Mark **Roget**, 1852)



Thesaurus

grand, *adj.* large, impressive, magnificent, stately, majestic; pretentious, ostentatious; elegant, lofty.

graph, *n.* diagram, chart, plot, PLAN; bar, circle, *etc.* graph.
graphic, *adj.* pictorial, descriptive; vivid, diagrammatic; delineative; picturesque. See REPRESENTATION,

graceless attractive. See BEAUTY, ELEGANCE.
graceless, *adj.* ungracious, tactless; inept, awkward, clumsy, ungainly; inelegant; sinful, corrupt. See UGLINESS, IRRELIGION, IMPENITENCE, INELEGANCE.
gracious, *adj.* gentle, courteous, tactful; kind, thoughtful, benign; affable, obliging; generous; charming, attractive. See BENEVOLENCE, COURTESY.
gradation, *n.* stage, DEGREE.
grade, *n.* level, quality, class; rank, standing; gradation, slope, tilt, slant. See OBLIQUITY, DEGREE.
gradual, *adj.* slow, progressive, moderate, leisurely. See DEGREE, SLOWNESS.
graduate, *n.* measure, beaker. *Slang,* grad. See DEGREE. —*v.* measure, classify, grade; calibrate; pass, commence. See ARRANGEMENT, MEASUREMENT, SCHOOL.
graffiti, *n.pl.* See WRITING.
graft, *v.* inoculate, bud, transplant, implant, join, crossbreed. See MIXTURE, INSERTION. —*n.* corruption, porkbarrel politics. See STEALING, IMPROBITY.
grain, *n.* fruit, cereal, seed, grist, kernel; TEXTURE, temper, TENDENCY; mite, speck, bit (see LITTLENESS).

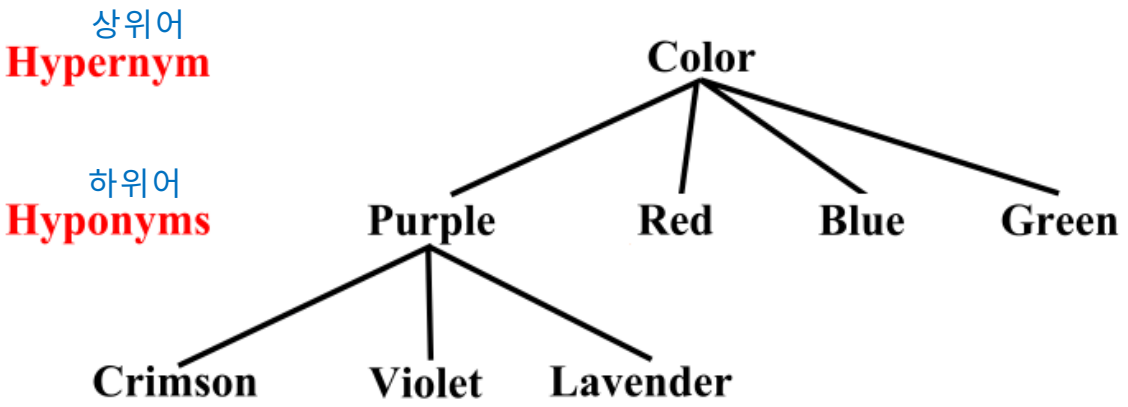
GRAMMAR

Nouns—**1,** grammar; accidence, syntax, analysis, synopsis, praxis, punctuation, syllabi[fi]cation; parts of speech; participle; article, noun, substantive, pronoun, verb, adjective, adverb, preposition, postposition, interjection, conjunction; particle; prefix, suffix, combining form, element; inflection, inflexion, case, declension, conjugation.
2, tense: present, past, preterit, future; imperfect, perfect, past perfect, pluperfect; progressive, *etc.*
3, mood, mode: infinitive, indicative, subjunctive, imperative.
4, sentence, paragraph, clause, phrase.
5, style; philology, language; phraseology, wording, rhetoric, diction. See SPEECH; WRITING.
Verbs—parse, analyze, diagram; punctuate; conjugate, decline, inflect.
Antonyms, see ERROR.

grand, *adj.* large, impressive, magnificent, stately, majestic; pretentious, ostentatious; elegant, lofty. See GREATNESS, OSTENTATION, REPUTE.
grandeur, *n.* GREATNESS, magnificence; show, ostentation; splendor, majesty; eminence, stateliness, loftiness. See IMPORTANCE.
grandfather, *n.* grandpa, grandsire; gaffer, old man. See ANCESTRY, AGE.
grandiose, *adj.* grand; stately, pompous, bombastic. See BOASTING, OSTENTATION.
grandmother, *n.* granny, grandma, nana, nanny; old woman. See ANCESTRY, AGE.
grandstand, *n.* bleachers. See VISION.
grant, *n.* gift, allotment, contribution. —*v.* bestow, give, yield; concede; permit; contribute. See GIVING, PERMISSION, CONSENT, RIGHTNESS, TRANSFER.
granular, *adj.* granulated, grainy, mealy, gritty, sandy. See POWDERINESS.
grapevine, *n., colloq.,* rumor; rumor mill, pipeline, a little bird. See INFORMATION.
graph, *n.* diagram, chart, plot, PLAN; bar, circle, *etc.* graph.
graphic, *adj.* pictorial, descriptive; vivid, diagrammatic; delineative; picturesque. See REPRESENTATION, DESCRIPTION.
grapple, *v.* seize, grasp, clutch, struggle, contend. See OPPOSITION, CONTENTION.

WordNet

- From Wikipedia,
  - A lexical database of semantic relations between words
  - WordNet links words into semantic relations including synonyms, hyponyms, meronyms and holonyms
    어떤 것의 부분을 나타내는 말, 전체를 나타내는 말
  - It is accessible to human users via a web browser, but most importantly, it is **machine readable** and its primary use is in NLP (Natural Language Processing) applications



WordNet

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: computer Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- **S:** (n) **computer, computing machine, computing device, data processor, electronic computer, information processing system (a machine for performing calculations automatically)**
  - *direct hyponym* / *full hyponym*
  - *part meronym*
  - *domain category*
  - *domain term category*
  - *direct hypernym* / *inherited hypernym* / *sister term*
  - *part holonym*
  - *derivationally related form*
- S: (n) calculator, reckoner, figurer, estimator, **computer** (an expert at calculation (or at operating calculating machines))

WordNet

- S: (n) **computer**, computing machine, computing device, data processor, electronic computer, information processing system (a machine for performing calculations automatically)
  - ***direct hyponym*** / *full hyponym*
    - S: (n) analog computer, analogue computer (a computer that represents information by variable quantities (e.g., positions or voltages))
    - S: (n) digital computer (a computer that represents information by numerical (binary) digits)
    - S: (n) home computer (a computer intended for use in the home)
    - S: (n) node, client, guest ((computer science) any computer that is hooked up to a computer network)
    - S: (n) number cruncher (a computer capable of performing a large number of mathematical operations per second)
    - S: (n) pari-mutuel machine, totalizer, totaliser, totalizator, totalisator (computer that registers bets and divides the total amount bet among those who won)
    - S: (n) predictor (a computer for controlling antiaircraft fire that computes the position of an aircraft at the instant of a shell's arrival)
    - S: (n) server, host ((computer science) a computer that provides client stations with access to files and printers as shared resources to a computer network)
    - S: (n) Turing machine (a hypothetical computer with an infinitely long memory tape)
    - S: (n) web site, website, internet site, site (a computer connected to the internet that maintains a series of web pages on the World Wide Web) *"the Israeli web site was damaged by hostile hackers"*

WordNet

- S: (n) **computer**, computing machine, computing device, data processor, electronic computer, information processing system (a machine for performing calculations automatically)
  - *direct hyponym* / *full hyponym*
  - ***part meronym***
    - S: (n) busbar, bus (an electrical conductor that makes a common connection between several circuits) *"the busbar in this computer can transmit data either way between any two components of the system"*
    - S: (n) cathode-ray tube, CRT (a vacuum tube in which a hot cathode emits a beam of electrons that pass through a high voltage anode and are focused or deflected before hitting a phosphorescent screen)
    - S: (n) central processing unit, CPU, C.P.U., central processor, processor, mainframe ((computer science) the part of a computer (a microprocessor chip) that does most of the data processing) *"the CPU and the memory form the central part of a computer to which the peripherals are attached"*
    - S: (n) chip, microchip, micro chip, silicon chip, microprocessor chip (electronic equipment consisting of a small crystal of a silicon semiconductor fabricated to carry out a number of electronic functions in an integrated circuit)
    - S: (n) computer accessory (an accessory for a computer) *"when you add in all the computer accessories you are going to need the computer gets pretty expensive"*
    - S: (n) computer circuit (a circuit that is part of a computer)
    - S: (n) data converter (converter for changing information from one code to another)
    - S: (n) disk cache (a cache that stores copies of frequently used disk sectors in random access memory (RAM) so they can be read without accessing the slower disk)

Relevance Feedback

설명 안들음!

- Idea

  컬렉션을 잘 모르는 경우 좋은 쿼리를 만들기 어려울 수 있다

  - It may be difficult to formulate a good query when you don't know the collection well.
  - Image search is a good example: 이미지 검색도 좋은 예시임
    Difficult to formulate a query, but easy to mark retrieved images relevant or nonrelevant.
- Users involved in IR process want to improve the final search results:

  초기 결과셋에 대해 관련성 피드백 받기

  - Users feedback on relevance of docs in initial set of results.
  - IR system refines the initial query to present better results.

    피드백으로 결과를 refine해 더 나은 결과 만들기

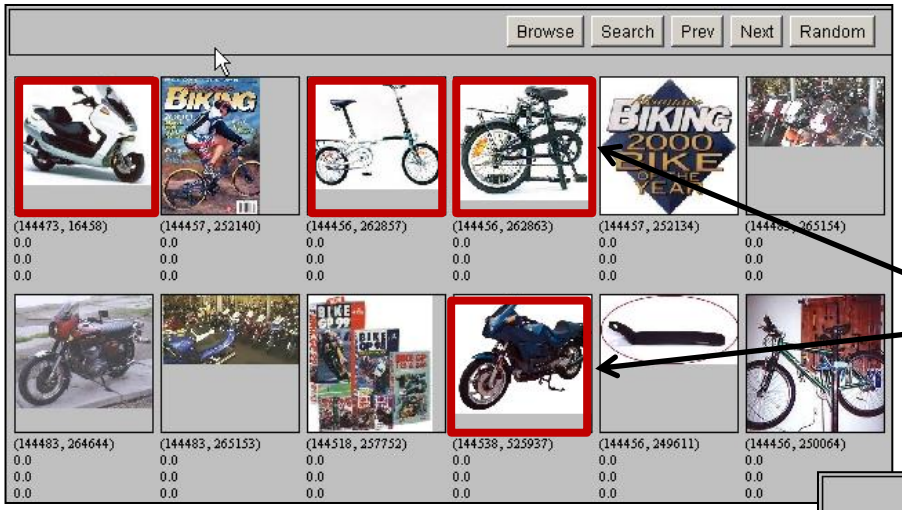
Relevance Feedback

- Basic Procedure of Relevance Feedback
  - User issues a (short, simple) query.
  - The system returns an initial set of retrieved results.
  - The user marks some results as ***relevant*** or ***non-relevant***.
  - The system computes a better representation of the information need based on feedback.
  - The system displays a **revised set of retrieved results**.
- The procedure can go through one or more iterations.
- ***ad hoc retrieval*** is regular retrieval without relevance feedback.

이 피드백을 받지 않는 검색 시스템은 "애드 혹"이라고 함

1. 사용자가 짧고 간단한 질의 생성
2. 시스템이 해당 질의에 대한 결과 반환
3. 사용자가 결과의 관련 여부를 평가
4. 시스템이 피드백 결과를 반영
5. 충분할 때까지 반복

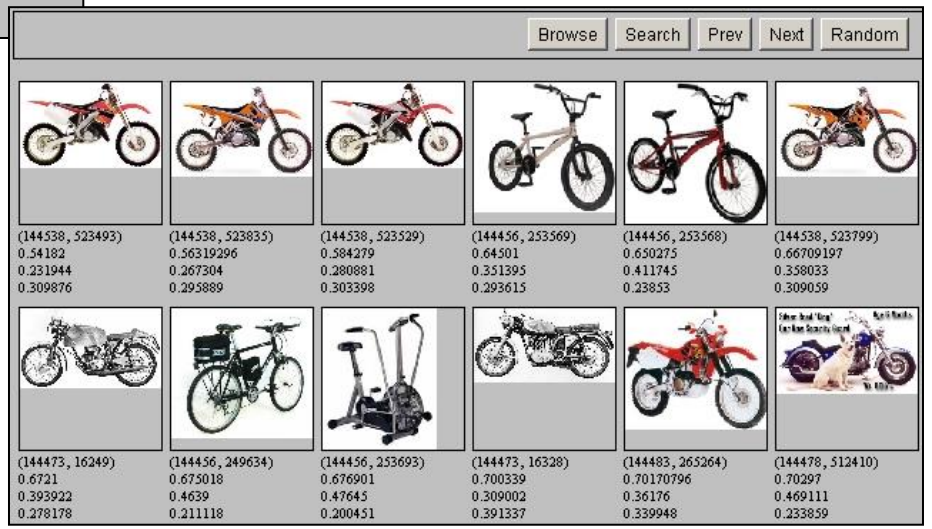
Relevance Feedback : Example 1



← initial search results

user feedback

improved search results →



Relevance Feedback : Example 2

Query: New space satellite applications

retrieved documents

relevance feedback

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

Improved results

1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
4. 0.493, 07/31/89, NASA Uses 'Warm' Superconductors For Fast Circuit

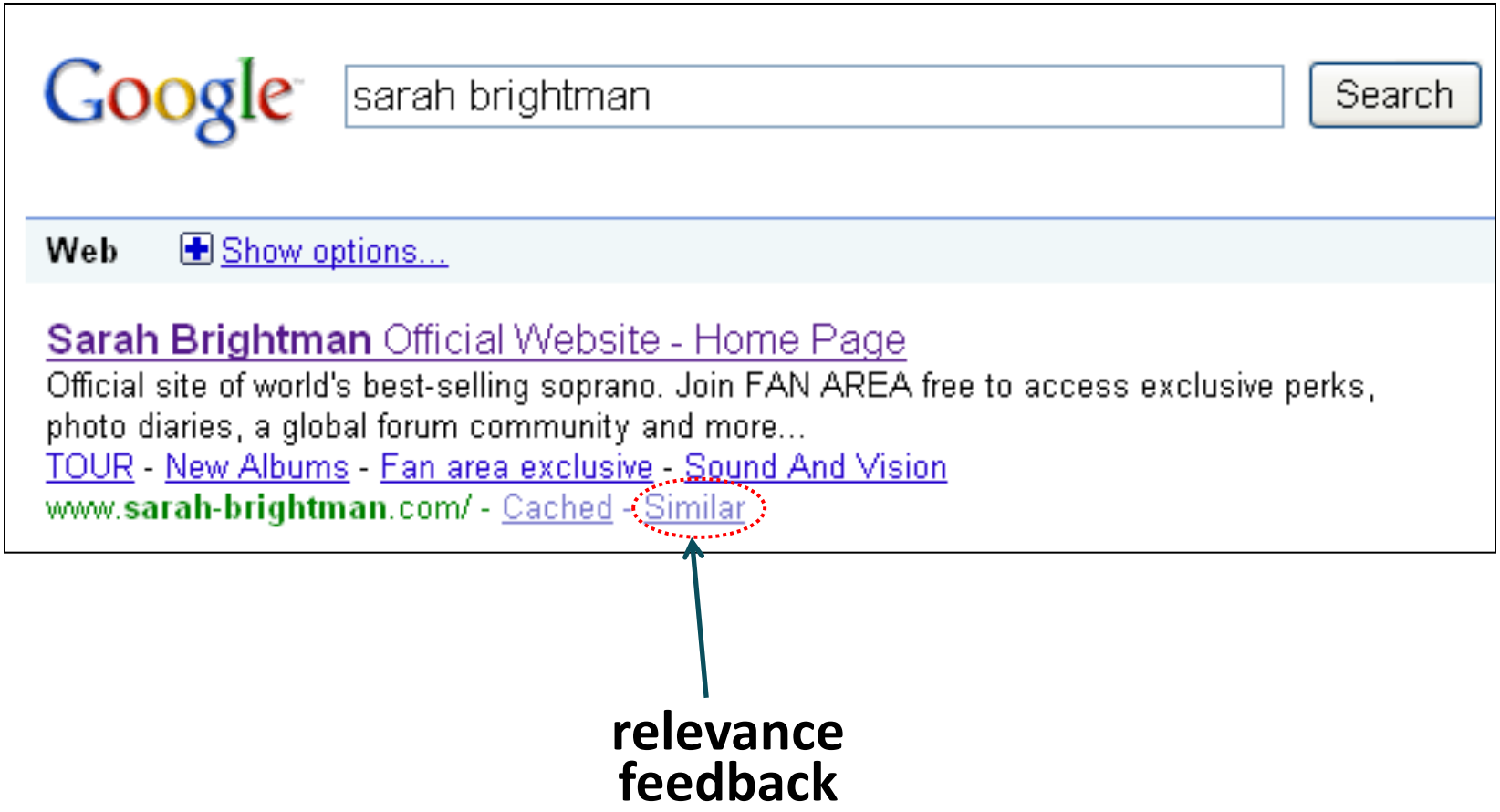
Relevance Feedback : Example 2

- Expanded query after relevance feedback

2.074	**new**	**15.12**	**space**
30.82	**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

Query: New space satellite applications

Relevance Feedback : Example 3



Local Methods

1. Key Concepts
2. Relevance feedback
3. Pseudo relevance feedback
4. Indirect relevance feedback
5. Evaluation of Relevance Feedback

Key Concept : Centroid

Key Concept : Centroid

- Definition

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

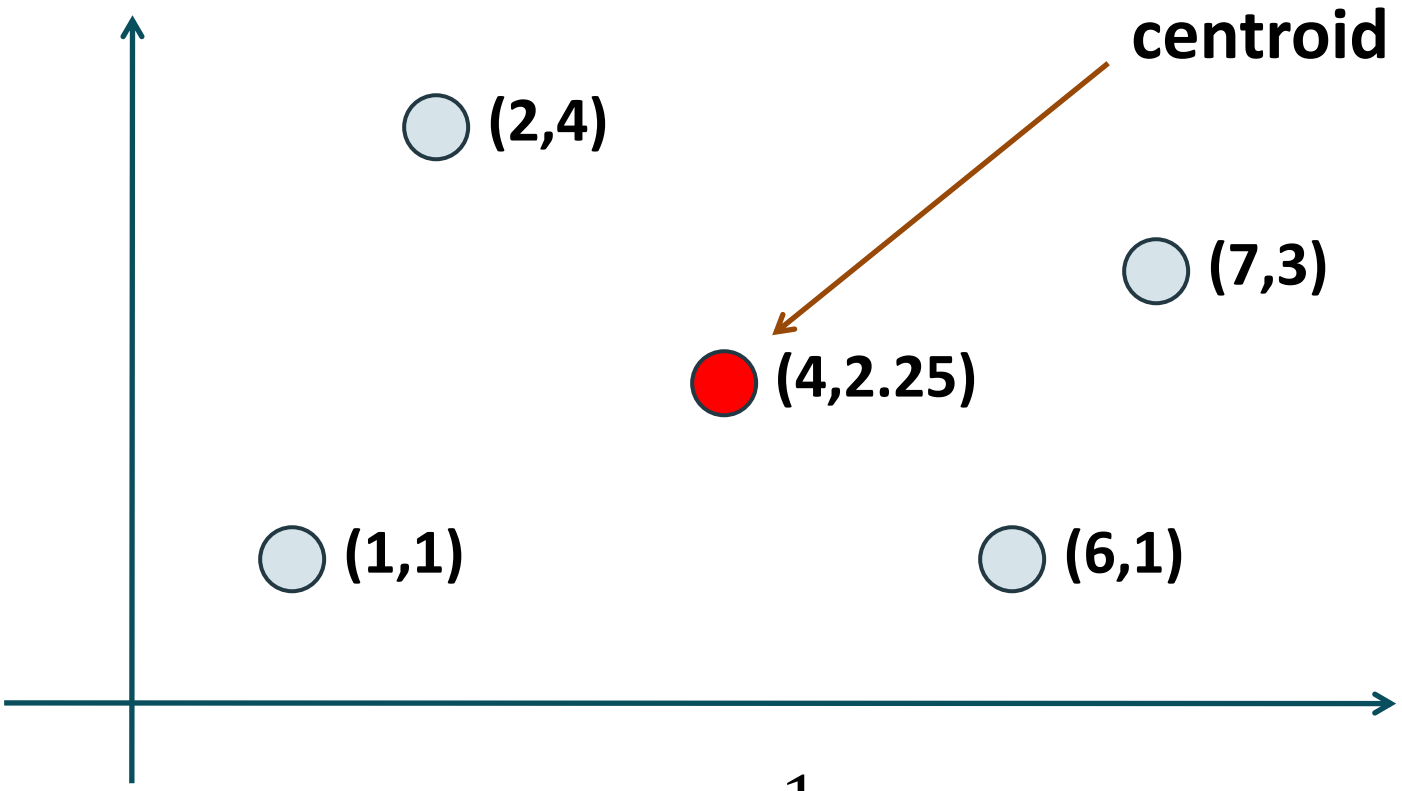
각 문서별 벡터의 같은 위치 요소를 다 더하고 그 개수로 나눔
그걸 벡터의 모든 요소에 대해 함
그 결과가 Centroid 값

where *C* is a set of documents. C는 문서 집합

- Recall that we represent documents as points in a high-dimensional vector space.
- The centroid is the center of mass of a set of points.

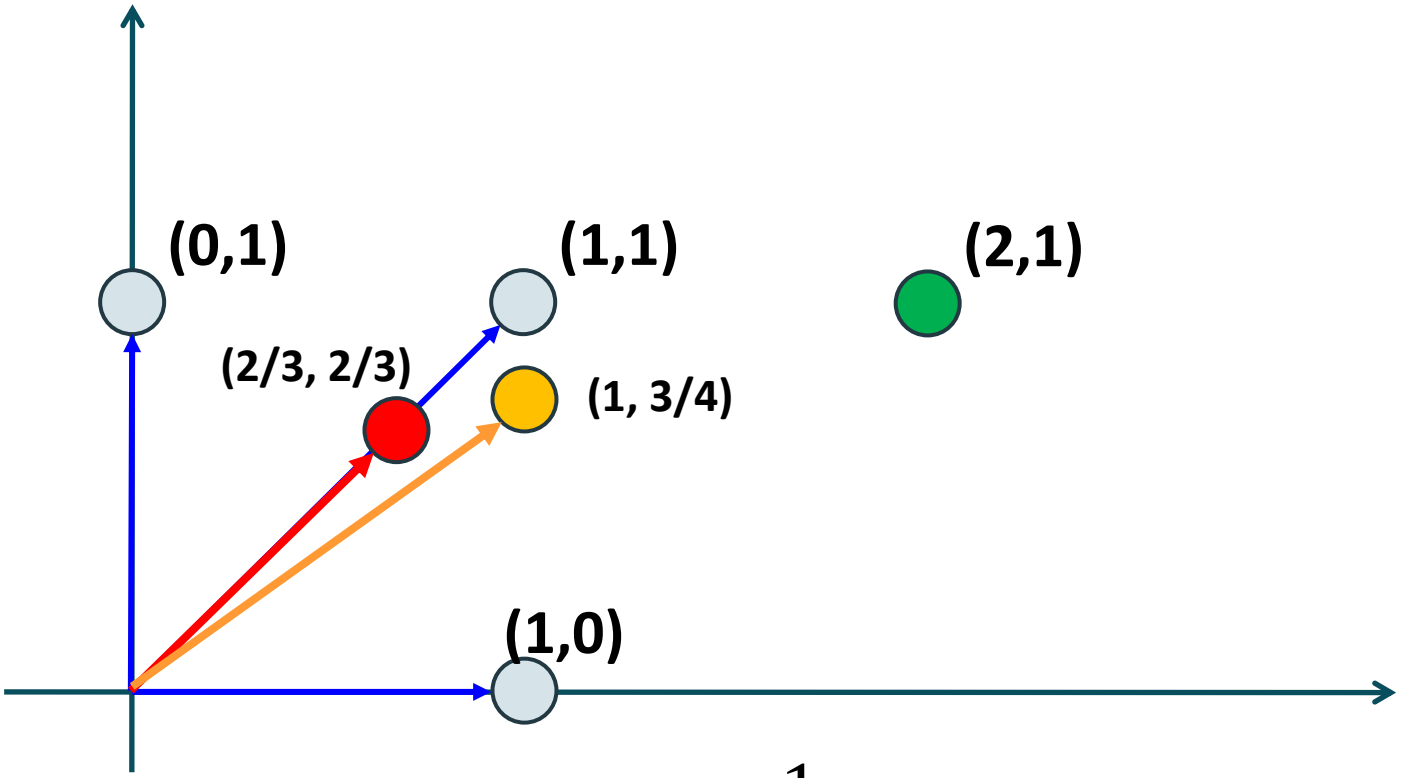
각 문서를 벡터공간에서 점으로 표시했을 때
그 점들의 무게중심이 centroid

Key Concept : Centroid



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

Key Concept : Centroid



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

Theoretically Optimal Query

- We want to find a query vector $\vec{q}$ that **maximizes similarity with relevant docs** while **minimizing similarity with non-relevant docs** *in the collection*.

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

where C_r is a set of relevant documents and C_{nr} is a set of non-relevant documents in the collection.

어떤 질의와 모든 문서에 대해 관련성 피드백을 다 받았다고 치면
그것으로 만들 수 있는 가장 최적의 질의는 위의 *argmax* 식을 통해 만들어진다
*argmax*란 뒤에 있는 식의 값을 가장 크게 만들 수 있는 *q*를 찾는 것
찾아야 할 대상은 보통 argmax 글자 밑에 써있다

Theoretically Optimal Query

- In a vector space model, the optimal query is:

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

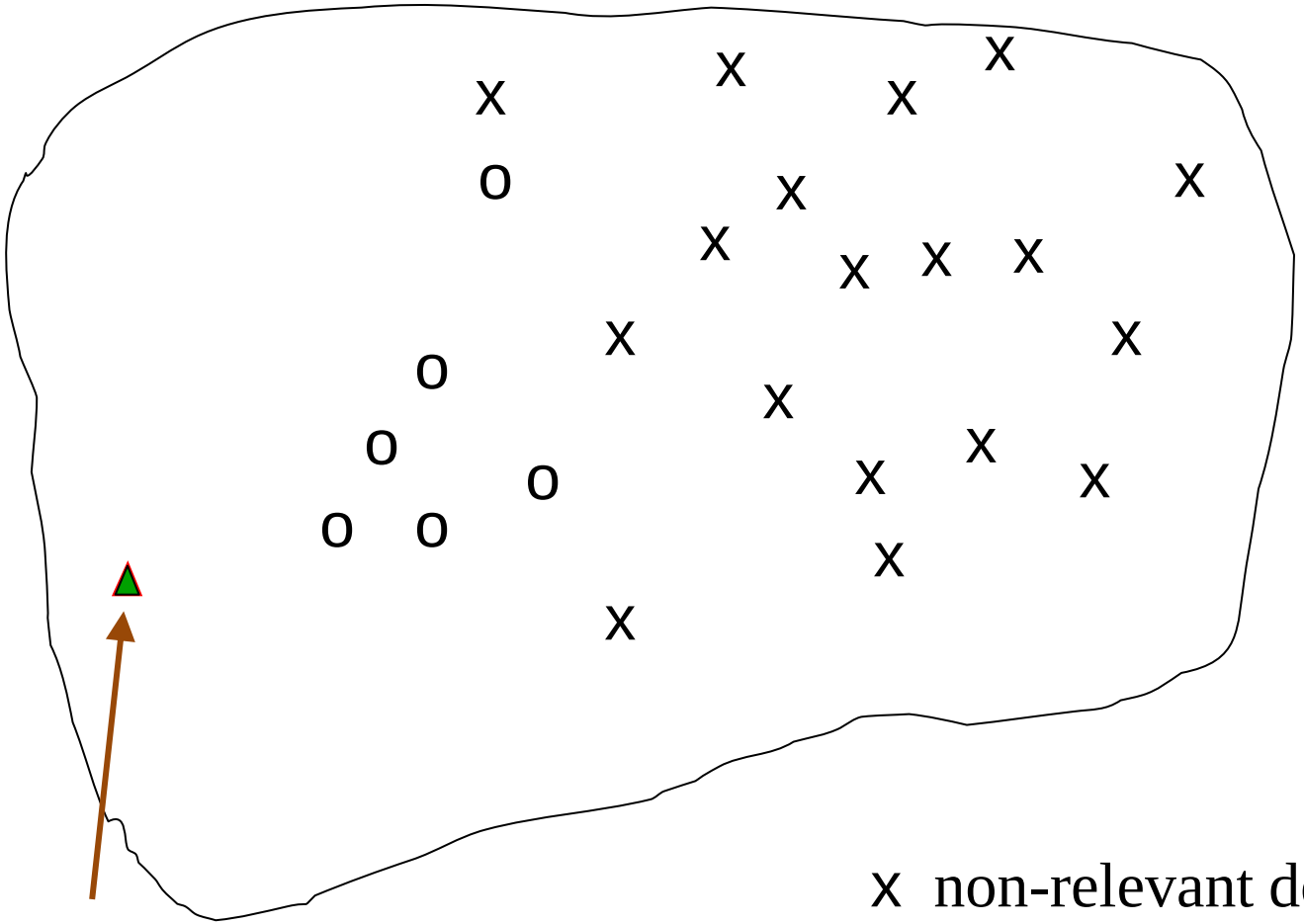
관련 문서의 centroid와 관련 없는 문서의 centroid 값의 차

  which is the vector difference between the centroids of the relevant and non-relevant document in the collection.

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

Theoretically Optimal Query

Relevance for all documents in the collection



Theoretically Optimal Query

- The problem of theoretically optimal query:
  - The **full set** of relevant and non-relevant documents is **not known in advance**. 모든 문서에 대한 관련성은 미리 알 수 없다
  - In fact, it is what we want to find!!! 그걸 찾아야 하는 거니까!
- In practice, we have a user query and **partial knowledge** of known relevant and non-relevant documents for the retrieved documents. 검색 결과에 대해 부분적 피드백만 받기
- We will use the partial knowledge to modify the query so that we can decide **pseudo optimal query**. 가 최적 질의를 만들기 위한 방법들
  - This is a relevance feedback mechanism.
  - **Rocchio Algorithm : Relevance Feedback Algorithm**

Rocchio Algorithm
Relevance Feedback Algorithm

Rocchio (1971) Algorithm

- Rocchio algorithm uses the vector space model to pick a relevance fed-back query.

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

where D_r = set of known relevant doc vectors

D_{nr} = set of known non-relevant doc vectors

q_0 = initial query vector

q_m = modified query vector

α,β,γ = weights, hand-chosen or set empirically

D는 전체 문서 중 사용자가 관련성을 피드백한 문서들. 이들의 centroid 값을 서로 빼고
초기 질의의 값을 더하면 처음에 비해 modify된 새 질의가 만들어짐
α, β, γ는 가중치 : 감으로 정하거나 실험적으로 정할 수 있다

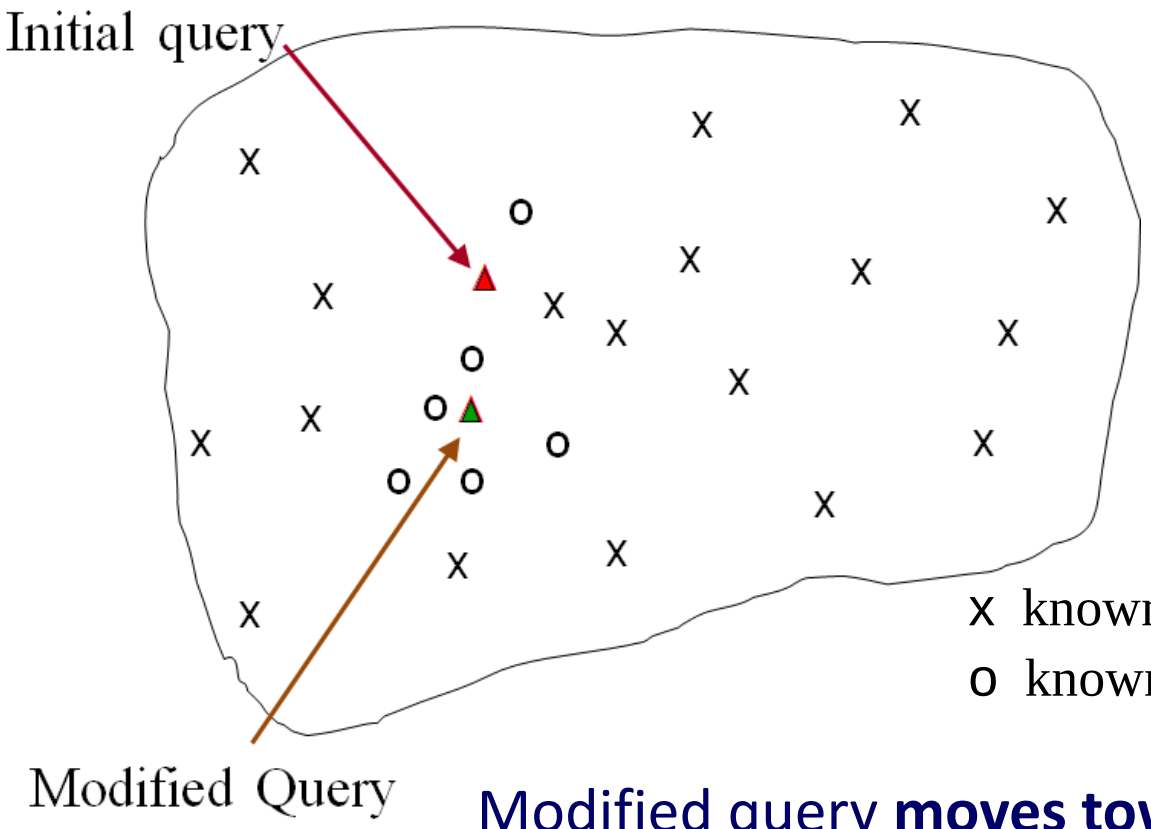
Rocchio (1971) Algorithm

- If we have a lot of judged documents, we want a higher β/γ. 만약 피드백을 많이 받았다면 β와 γ는 크게 줘도 된다.
- Positive feedback is more valuable than negative feedback, so, set $\beta > \gamma$. 관련 있는 문서를 찾는 게 더 가치있기 때문에 보통 β를 γ보다 크게 준다
  - e.g. $\beta = 0.75$, $\gamma = 0.25$
- Many systems only allow positive feedback (γ=0).

보통 사용자에게 '관련 있음' 피드백만 받기 때문에 γ = 0으로 둔다

Rocchio (1971) Algorithm

Modify query with partial knowledge



Modified query **moves toward** the centroid of relevant documents and **away from** the centroid of nonrelevant documents.

Optimal Query vs Rocchio Algorithm

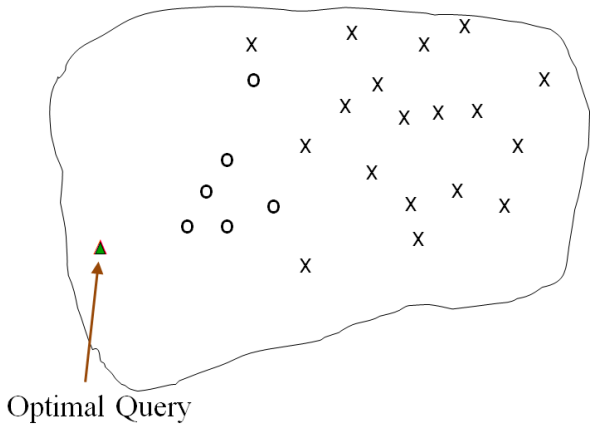
- Optimal query 모든 문서에 대해 관련성을 알고있음

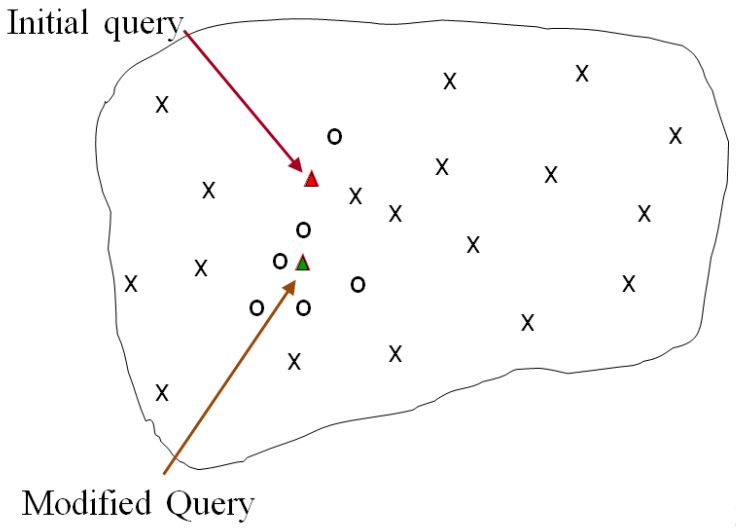
$$\vec{q}_{opt} = \frac{1}{|C_r|}\sum_{\vec{d}_j \in C_r}\vec{d}_j - \frac{1}{|C_{nr}|}\sum_{\vec{d}_j \in C_{nr}}\vec{d}_j$$

- Rocchio algorithm 일부 문서에 대한 관련성만 알고 있음

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

Why $\vec{q}_0$ is added here?





Rocchio (1971) Algorithm

- In summary,
  - We can modify the query based on relevance feedback and search again with the modified query.
  - Modify the query only with the docs that were marked relevant or non-relevant.
  - 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 on the search.

최적의 질의를 찾는 이유는 재현율을 높이기 위해서다
대신 시간을 좀 들여야 함

Pseudo Relevance Feedback

Pseudo Relevance Feedback

- Also known as ***blind 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).

쿼리 결과에 대해 사용자가 많이 클릭한 상위 k개가 '관련 있음' 문서라 전제하고 피드백으로 반영

Pseudo Relevance Feedback

- Works very well on average. 평균적으로 효과가 있는 편
- But can go horribly wrong for some queries. 가끔 이상한 게 나옴
- Several iterations can cause query drift.
  - Assume that the initial query is “copper mines”.
  - The top *k* documents are all about mines in Chile.
  - Then, the modified query may drift in the direction of documents on Chile.

이상한 결과의 예시:
1. 사용자가 "구리 광산" 검색
2. 상위 k개 문서 중 칠레의 광산에 대한 문서가 있었음
3. 쿼리가 칠레에 대한 문서를 반환하도록 modify됨 : 칠레 자체는 원래 질의와 전혀 관련이 없음

Indirect Relevance Feedback

Indirect Relevance Feedback

- Also known as ***implicit relevance feedback***.
  간접적/묵시적 관련성 피드백
- Idea:
  - Assumption is that document summaries displayed in results lists are indicative of the relevance of these documents.
  - Then, clicks on links indicate that the page was likely relevant to the query.
  - The data about click rates on pages were gathered globally, rather than being user or query specific.
- Thus, **rank documents higher** that users look at more often.
  사용자는 검색 결과가 원하는 정보일 것 같아서 클릭하기도 하지만
  여러 가지 이유로 원하는 정보가 아닌데도 클릭하는 경우도 많다.
  그래서 사용자 개개인의 클릭 정보는 그다지 유효하지 않지만
  다수 사용자의 클릭을 유독 많이 받는 문서들은 질의와 관련이 있다고 볼 수 있다.
  고로 질의를 수정하지는 않고, 사용자의 클릭 정보를 이용해 더 관련있을 것으로 보이는 문서를 노출한다.

Indirect Relevance Feedback

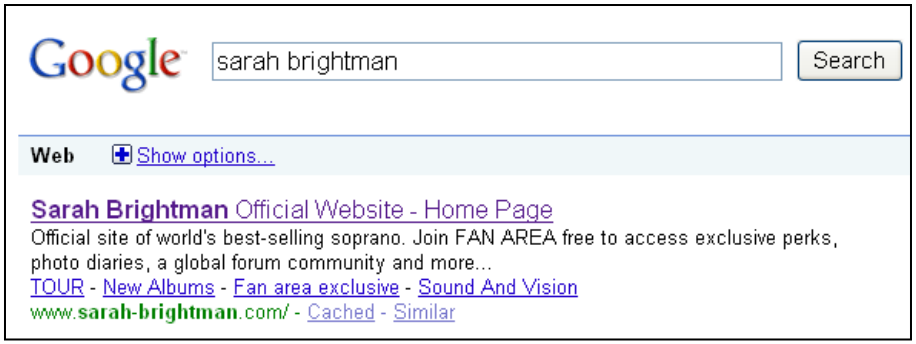
- This is the general area of **click stream mining**.
- Similar approach is used in ranking the advertisement that match a web search query.
- Less reliable than explicit feedback, but is more useful than pseudo relevance feedback.
- Easy to collect indirect feedback in large quantities, esp. on a web search engine.

이 방법은 클릭 스트림 마이닝이라고 부른다
명시적 피드백보다는 덜 믿음직해도 더 유용하다

and more ...

Relevance Feedback on the Web

- Some search engines offer a *similar*/*related* pages feature (this is a trivial form of relevance feedback).
  - Google (similar pages)
  - Altavista (more pages)



- Some don't because it's hard to explain to average users.
  - Alltheweb (closed in 2011)
  - Yahoo

Excite Relevance Feedback

- *Excite* initially had true relevance feedback, but dropped it due to lack of use. Excite라는 사이트에 검색 결과 피드백 기능이 있었는데, 사용자들이 너무 안 써서 없어졌다
- Research by Spink et al. 2000
  - Relevance feedback improved results about 2/3 of the time. 관련성 피드백은 보통 결과의 2/3정도를 개선했다.
  - Only about 4% of query sessions from a user used relevance feedback option. 사용자의 4%정도만이 피드백 기능을 사용
    - Expressed as "More like this" link next to each result.
  - But about 70% of users only looked at first page of results and didn't pursue things further.

    70%는 검색 결과가 마음에 안들면 다시 검색을 하지 검색 결과의 다른 페이지를 보려고 하지 않음

Problems of Relevance Feedback

- Users are often reluctant to provide explicit feedback.
- It's often harder to understand why a particular document was retrieved after applying relevance feedback.

Problems of Relevance Feedback

- Long (modified) queries are inefficient for typical IR engine. 확장된 쿼리는 비효율적이다
  - Long response times for user. 결과 제공이 느려짐(계산량 증가)
  - High computing cost for retrieval system. 컴퓨팅 자원도 많이 씀
  - Partial solution 부분적 해결 방법은 있지만 그래도 원래 쿼리보다는 비효율적임
    reweight certain prominent terms only
    e.g. reweight only top 20 terms by term frequency

Query: New space satellite applications

2.074	**new**	15.12	**space**
30.82	**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

Evaluation of Relevance Feedback

관련성 피드백 평가하기

Evaluation of Relevance Feedback

- Empirically, one round of relevance feedback is often very useful. Two rounds is sometimes marginally useful.
- **Evaluation Strategy 1**

최초 질의를 피드백해서 한 번 수정하면 보통 결과가 크게 개선되지만
두 번째 수정은 별로 큰 개선이 없으니 한번만 수정해도 된다

  - Use q_0 and compute precision and 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.

평가 방법 1 : 초기 쿼리와 이후 수정을 거친 쿼리들의 결과를 비교하는 것
→ 피드백을 받으면 개선되는 게 당연하니까 부적절한 평가 방법이다.

시험 봤는데 맞은 문제 포함 재시험보고 점수 비교하는거랑 같은 식이다

Evaluation of Relevance Feedback

Query: New space satellite applications

retrieved documents

relevance feedback

\+ 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

improved results

1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
4. 0.493, 07/31/89, NASA Uses 'Warm' Superconductors For Fast Circuit

Evaluation of Relevance Feedback

- **Evaluation Strategy 2**
  - Assess only the documents *not* rated by the user in the first round (*residual collection*).
  - Measures are usually lower than for original query because a fair proportion of them have been judged by the user in the first round.
  - So, can be used to compare relative performance of variant relevance feedback methods.
  - Cannot be used to compare performance with and without relevance feedback because the collection size and the number of relevant documents changes.

평가 방법 2 : 피드백을 받긴 하는데, 피드백받은 문서는 다 빼고 검색함.
시험 보고 맞은 문제 빼고 재시험보는 방식
여러 가지 시스템의 상대적인 비교에는 쓸 수 있지만
피드백 유무에 따른 성능 비교에는 부적절한 방법임

Evaluation of Relevance Feedback

- **Evaluation Strategy 3**
  - Most satisfactory strategy.
  - Use two collections each with their own relevance assessments.
    - q_0 and user feedback from first collection
    - q_0 and q_m run on second collection to measure the performance.

평가 방법 3 : 문서 집합을 골고루 반으로 나눔
첫 번째 문서 집합에 대해 검색 후 피드백
두 번째 문서 집합에 대해 개선된 쿼리로 검색

Evaluation of Relevance Feedback

- Relevance Feedback vs User's revision of query

  관련성 피드백과 사용자가 직접 재검색하는 걸 비교해보자

  - User revises and resubmits query.
    Users may prefer revision/resubmission to judging relevance of documents.

어느쪽을 쓰든 사용자는 시간을 들여야 한다

- True evaluation of usefulness must compare to other methods taking the same amount of time.
- There is no clear evidence that relevance feedback is the "best use" of the user's time.

결과만 볼 게 아니라 사용자가 들인 시간도 고려해서 평가해야 한다
관련성 피드백이 무조건 좋은 방법이라는 증거는 없다

Global Methods

사전을 이용한 질의 확장

1. Query Expansion 사전이 준비되어 있을 때
2. Automatic Thesaurus Generation

사전이 없을 때

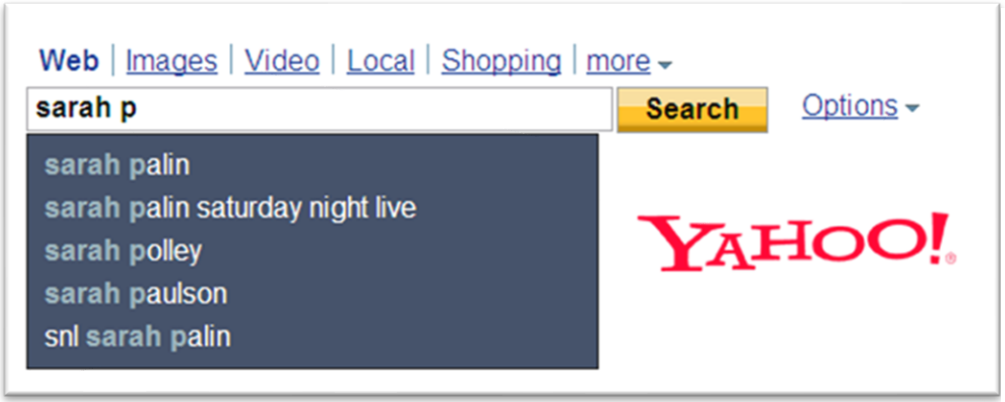
1. Query Expansion

- **Relevance Feedback**
  - Users give additional input (relevant/non-relevant) **on documents**, which is used to reweight terms in the documents.
- **Query Expansion** 쿼리 확장 : 쿼리에 단어를 추가함
  - Users give additional input **on query words or phrases**.
  - Some search engines suggest related query terms in response to a query.
  - **Thesaurus** is often used in expanding queries.

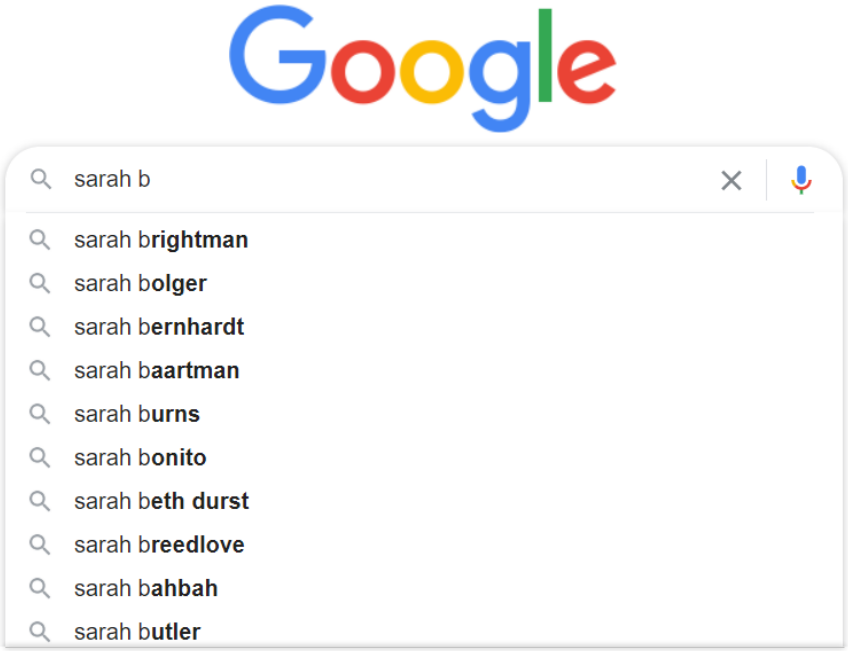
    쿼리 확장을 사람이 하긴 하는데, 추가할 단어를 추천해줌
    여기에 사전을 활용하기도 한다

Query Assist

- Suggest related terms based on **query log mining**.
  - Recommend frequent recent queries that contain partial string typed by user.
  - **Users choose** one from the suggestions.

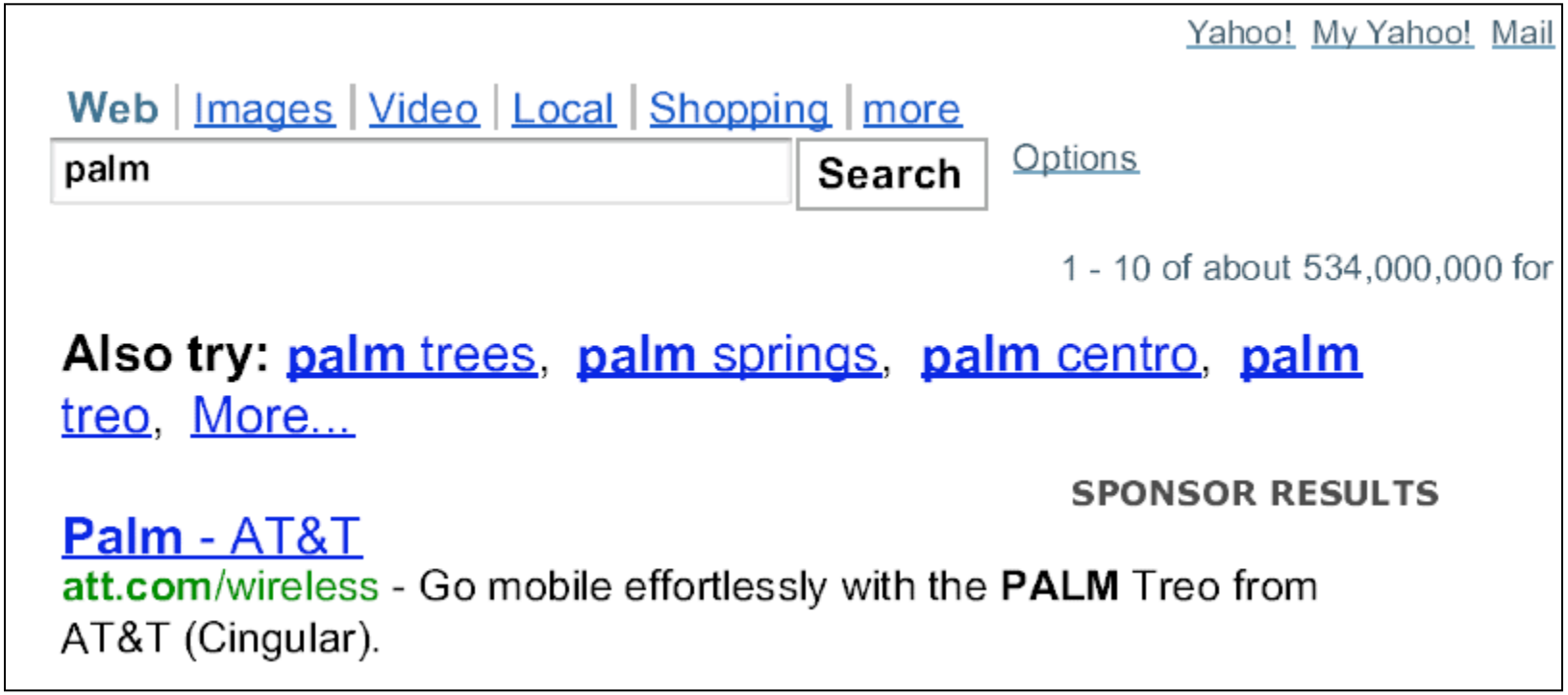


사용자들의 검색어 로그를 모아뒀다가 검색어 추천에 씀



Query Assist

- Suggest related terms based on **query log mining**.
  - Recommend frequent recent queries that contain partial string typed by user. 검색 후에 연관검색어를 주는 방법도 있다
  - **Users choose** one from the suggestions.



Thesaurus-based Query Expansion

- For each term *t* in a query, expand the query with synonyms and related words of *t* from the thesaurus.
  - feline → feline cat
  - Weight may be added to expanded terms, but less than that for original query terms.
- Generally query expansion increases recall.
  - Thus, widely used in many science and engineering fields.
  - However, precision may decrease significantly.

유의어 사전을 이용해 쿼리 확장하기
시스템이 알아서 확장함
여기서 추가되는 단어에 가중치를 줄 수도 있는데 원래 질의에 있는 단어보다는 작게 줘야 함

이렇게 확장하면 보통 재현율은 높아지는데 정밀도가 떨어질 수 있다

Thesaurus-based Query Expansion

- Use manual thesaurus 유의어 사전은 보통 수작업으로 만들어진 것이다
  - Thesaurus is a dictionary containing terms such as preferred terms, synonyms, broader terms, narrower terms, …
  - e.g. MedLine : physician, syn: doc, doctor, MD, medico

User query: cancer

PubMed query: ("neoplasms"[TIAB] NOT Medline[SB]) OR "neoplasms"[MeSH Terms] OR cancer[Text Word]

PubMed does automatic query expansion.

Thesaurus-based Query Expansion

- Drawbacks of manual thesaurus
  - High cost of manually producing a thesaurus.
  - Too general manual thesaurus has little coverage of rich domain-specific vocabularies.

→ automatic generation is preferred.

수제 유의어 사전의 단점
만들기 힘듦
보통 특정 도메인(분야)에 한정된 용어에 대해서는 잘 나타내기 힘들다(전문 사전이 아닌 이상)

→ 자동 사전 생성이 필요하다

2. Automatic Thesaurus Generation

- Automatic thesaurus is generated based on the co-occurrence statistics over a collection of documents in a domain.

- Fundamental notion
  - Capture **similarity between two words** from the collection.

자동 유의어 사전 만들기
두 단어가 문서에 함께 등장하는(co-occurrence) 통계를 이용한다

2. Automatic Thesaurus Generation

- Definition 1 두 단어가 한 문서/문단/문장 안에서 같이 등장하면 의미적으로 유사한 것이다

  Two words are similar if they co-occur in the same document, paragraph or sentence.

Query expansion is often effective in increasing **recall**. However, **query expansion** may also significantly decrease **precision**. In general, a domain-specific **thesaurus** is required.

query expansion	**recall precision thesaurus**

두 단어가 같이 등장한 수가 한번이면 관련이 있다고 보기 힘들지만
수많은 문서에서 계속 같이 나오면 관련이 있다고 볼 수 있다

2. Automatic Thesaurus Generation

자동 생성 유의어 사전 예시 : 가끔 잘못 만들어진 것도 있긴 함(keeping)

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

lithographs : 석판화

2. Automatic Thesaurus Generation

- Definition 2 문법적으로 비슷한 위치에 나왔을 때 그 단어들은 유사하다

  Two words are similar if they occur in a given grammatical relation with the same words.

  They harvest apples. He peels apples. She eat apples.
  Jane harvests pears. Tom peels pears. They eat pears.

  Apples and Pears must be similar.

- Co-occurrence based thesaurus is more robust, grammatical relation based thesaurus is more accurate.

2. Automatic Thesaurus Generation

- Quality of word associations is usually a problem.
  - Term ambiguity may introduce statistically correlated but semantically irrelevant terms. 단어의 의미가 모호해서 여러 문제가 생길 수 있음
  - **False positives**: words deemed similar that are not.
  - **False negatives**: words deemed dissimilar that are similar.
- Bad News
  Since terms are highly correlated anyway, expansion may not retrieve many additional documents.

나쁜 소식
서로 같이 등장한 단어들을 이용해 유의어 사전을 만들었기 때문에
질의를 확장해도 검색 결과는 크게 달라지지 않을 수 있다(어차피 같은 문서에 나온 단어들이니까)