웹 검색의 과거와 링크 분석

웹 검색엔진의 과거

Timeline		
Note: "Launch" refers only to web availability of original crawl-based web search engine results.		
Year	**Engine**	**Event**
1993	Aliweb	Launch
1994	WebCrawler	Launch
	Infoseek	Launch
	Lycos	Launch
1995	AltaVista	Launch (part of DEC)
	Magellan	Launch (The McKinley Group)
	Excite	Launch
	SAPO	Launch
1996	Dogpile	Launch
	Inktomi	Founded
	HotBot	Founded
	Ask Jeeves	Founded
1997	Northern Light	Launch
1998	Google	Launch
1999	AlltheWeb	Launch
	Naver	Launch
	Teoma	Founded
	Vivisimo	Founded
2000	Baidu	Founded
2003	Info.com	Launch
2004	Yahoo! Search	Final launch
	A9.com	Launch
2005	MSN Search	Final launch
	Ask.com	Launch
	GoodSearch	Launch
2006	wikiseek	Founded
	Quaero	Founded
	Ask.com	Launch
	Live Search	Launch
	ChaCha	Beta Launch
	Guruji.com	Beta Launch
2007	wikiseek	Launched
	AskWiki	Launched

출처 : http://en.wikipedia.org/wiki/Web_search_engine

웹 검색엔진의 시장 점유율

Current market share

출처 : http://en.wikipedia.org/wiki/Web_search_engine

Most popular search engines worldwide, Dec. 2007[6][*not in citation given*]		
Company	Millions of searches	Relative market share
Google	28,454	46.47%
Yahoo!	10,505	17.16%
Baidu	8,428	13.76%
Microsoft	7,880	12.87%
NHN	2,882	4.71%
eBay	2,428	3.9%
Time Warner (includes AOL)	1,062	1.6%
Ask.com and related	728	1.1%
Yandex	566	0.9%
Alibaba.com	531	0.8%
Total	**61,221**	**100.0%**

* 네이버가 국내 검색시장의 약 **70%** 점유

웹 검색엔진의 과거

* "The next generation Web Search and the demise of the classic IR model", A. Broder, Yahoo Research, 2007 참조

- **1995 – 1997: 1 세대 – use only "on page" text data**
  - 웹문서내의 데이터/단어 만을 이용해 사용자 질의에 대답
  - 웹문서내의 단어 빈도수, 단어들의 위치를 참조
  - Altavista, Lycos, Excite, Infoseek, Inktomi, WebCrawler, Open Text, Hotbot
  - 돈을 받고 높은 검색 결과 순위에 올려줌: Goto.com → Overture.com → Yahoo

- **1998 – 2003? : 2 세대 – 페이지 밖의 정보, web-specific data 활용**
  - 웹문서들간의 Link (connectivity) 분석을 통해 웹문서의 중요순위를 매김
  - 구글이 대표적. 현재는 대부분의 검색엔진이 link 데이터 활용
  - 구글은 검색결과의 순위와 광고를 분리 : 광고란을 검색 옆에 별도로 두어 광고비를 많이 받아도 검색결과의 높은 순위에 올려주지 않음
  - 이전의 검색엔진들을 완전히 제압
  - Click-through data (What results people click on)
  - Anchor-text (How people refer to this page)
  - 이 시기 Goto/Overture 의 연매출이 $10억에 이르렀음
  - Yahoo가 Overture와 Inktomi를 매입

- **2004 – 현재: 3 세대 – Vertical Search, 통합검색, Universal search**
  - 웹문서, 이미지, 비디오, 뉴스, 블로그, 쇼핑, 지도, 책 등 다양한 컨텐츠 종류들을 통합하여 고객에게 보여줌.
  - 다양한 컨텐츠 종류와 주제에 대한 검색을 특정 컨텐츠 종류와 주제에 특화된 vertical search engine을 활용
  - 동시에 informational, navigational, and transactional 정보를 취합해 보여줌.
  - 검색어에 따라 검색결과의 보여지는 패턴이 다름.
  - 우리나라의 네이버 지식인과 통합검색이 선도적인 서비스임.

http://en.wikipedia.org/wiki/List_of_search_engines 에 가면 많은 검색엔진이 종류별로 분류되어 있음.

현재 웹 검색 결과:


NAVER
광우병
통합검색
지식iN
광우병 2008.05.10
이명박,광우병 2008.05.02
광우병쇠고기 2008.05.05
이명박대통령이 5월 1일부터 광우병걸린 소를...
광우병에대해서자세하게알려주세요 2008.05.02
연관검색어
광우병 뜻
광우병 동영상
광우병 사위
광우병미란
광우병 원인
광우병 증상
광우병 만화
pd수첩 광우병
인간광우병
pd수첩


NAVER
알렉스
스폰서광고
Alex


Google
알렉스
검색
웹문서
알렉스에 대한 뉴스 검색결과
'일밤' 시청률 1...'우리 결혼했어요' 알렉스-신애 공백 여파? - 12시간 전
알렉스
cyworld.nate.com/al_goon - 유사한 페이지
블로그
알렉스, '푸른밤' DJ 5월말 투입
보미와씨 - kr.blog.yahoo ...
2008년 5월 10일
신애...그리고...알렉스...
23시간 전
알렉스와 신애 진짜 연인 아니야 ...
좋은뉴스 (news.isegye.com)
2008년 5월 10일
알렉스 / 최문
pain.egloos.com
2008년 5월 10일
블로그 검색결과 더보기 »
이미지

검색결과 클릭율 및 시간

http://www.seopedia.org/wp-content/uploads/2006/10/click-distribution-serp.jpg

처음 몇 개의 검색결과가 대부분의 클릭을 차지하지만, **long tail** 특성도 함께 보인다; **Power law** 특성?

미래의 웹 검색엔진은?

Answering “the need behind the query”

- **개인 경험 & implicit 경험 및 지식 이용**
  - 개인적 취향, 검색 패턴
  - 유저의 검색행동 자동 취합 & 피드백 적용, 사용자들의 참여 유도
- **Intelligent**
  - 고객이 검색하고자 하는 진짜 목적/의도를 파악해 답을 제공
  - 때로는 “long tail” 을 활용한 “비상식적” 인 측면도 고려
  - 적절한 기술을 이용한 recommendation과 논리적 추론 결과 제공
  - 직관적인 UI를 통해 고객에게 빨리 답을 가르쳐 줌
- **Social Search, 사용자 참여 검색, AB search (일명, 알바 검색 ^v^)**
- **Hard & soft matches**
- **Context 정보 활용 (위치, 시간, 환경 등)**
- **Pervasive but invisible**
- **Always connected**

웹 검색의 활용-1 : 출처: A taxonomy of web search, 2002, A. Broder, IBM Research

- 정보 취득 – **want to learn about something (~39%)**
  - "알렉스 신애" 가 왜 요즈음 인기지?
  - **Users want to learn about a topic**
  - **They want to explore relevant pages**

- 내비게이션 – **want to go to that page (~25%)**
  - "현대자동차" 웹페이지가 어디지?
  - **Users want to visit a particular Web site or page**
  - **They already know what they want**

- Transactional – **want to do something (web-mediated) (~36%)**
  - **Access a service**
  - 다운로드
  - 쇼핑

✓ "Web Search Challenges and Progress, Junghoo Cho, 2007" 에 의하면 현재 가장 많은 검색요구는 웹페이지를 찾는 Navigation 이라고 함. 따라서 이 요구를 가장 잘 대응하는 것이 검색엔진의 목표임.

사람들은 웹검색에서 무엇을 찾으려 하지?

출처: **Determining the User Intent of Web Search Engine Queries,** Bernard J. Jansen, Danielle L. Booth, Amanda Spink, http://www2007.org/posters/poster989.pdf,

- 방법 : 3개의 검색엔진에 질의된 500백만 이상의 검색 질의를 활용
- 사용자의 질의를 3가지 카테고리로 나눔

1. 네비게이션형 질의

- queries containing company/business/organization/people names
- queries containing domains suffixes
- queries with “web” as the source
- queries length (i.e., number of terms in query) less than 3
- searcher viewing the first search engine results page

2. 정보형 질의

- uses question words (i.e., “ways to,” “how to,” “what is”, etc.)
- queries with natural language terms
- queries containing informational terms (e.g., list, playlist, etc.)
- queries that were beyond the first query submitted
- queries where the searcher viewed multiple results pages
- queries length (i.e., number of terms in a query) greater than 2
- queries that do not meet criteria for navigational or transactional

3. 트랜즈액션형 질의

- queries containing terms related to movies, songs, lyrics, recipes, images, humor, and porn
- queries with “obtaining” terms (e.g., lyrics, recipes, etc.)
- queries with “download” terms (e.g., download, software, etc.)
- queries relating to image, audio, or video collections
- queries with “audio”, “images”, or “video” as the source
- queries with “entertainment” terms (pictures, games, etc.)
- queries with “interact” terms (e.g., buy, chat, etc.)
- queries with movies, songs, lyrics, images, and multimedia or compression file extensions (jpeg, zip, etc.)

Table 1. Results from Automatic Classification of Queries

Classification	Occurrences	%
Informational	1,228,427	80.6%
Navigational	155,628	10.2%
Transactional	139,738	9.2%
	1,523,793	100.0%

- 앞의 Broder, 2002 결과와 다름

Link Analysis of the Web

자료 :

1. "Mining the Link Structure of the World Wide Web (1999)" - S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins

2. "Authoritative Sources in a Hyperlinked Environment (1997)" - Jon M. Kleinberg

* 위 저자들의 이름을 잘 기억하라. 웹 공부하다 보면 많이 나옴.

검색엔진의 문제점:

- Search engines are index-based and limited to keyword searches.
- Keyword search, even very narrowed, not always return expected results.
- Problem with correctness of results or too wide range.
- 페이지 내용 중에 그 페이지를 잘 설명하는 키워드/단어를 갖고 있지 않는 경우가 많다

  - "현대자동차" 홈페이지에 자동차 제작 회사라는 말이 많이 나오지 않는다

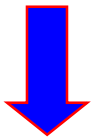
  - 구글, 네이버에 "검색엔진" 이라는 말이 많이 나오지 않는다

목표 :

- Design algorithms for mining link information.
- Develop techniques that take advantage of social organisation of the Web.
- Effectively find authorative pages in some field.
- Point knowledge hubs in some field.

HITS algorithm 개관

- ✓ 페이지-*p* 의 저자가 페이지내에 페이지-*q* 로의 링크를 포함했으면 페이지-q에 **authority**를 부여했다고 할 수 있다.
- ✓ 많은 수의 링크들은 단순히 웹 내비게이션용으로 만들어졌다
- ✓ 많은 수의 링크들은 광고성이다



- Computes hubs and authorities for search topic
  - **Authority** : In link를 많이 받는 페이지. Page rank가 높은 페이지.
  - **Hub** : Authority가 큰 곳으로 많은 Out-link를 보내는 페이지.
  - Certain natural type of equilibrium exists between hubs and authorities in the graph dened by the link structure
- Components
  - Sampling : It is processed on a small subset of 'relevant' documents, not all documents as was the case with PageRank
    - focused collection of pages likely to have many relavant authories
  - weight-propagation : It computes two scores per document, hub and authority
    - determine weights of hubs and authorities in iterative process

HITS algorithm

- Web representation - directed graph
- 알고리즘을 적용할 웹의 일부분 sub graph를 구한다.
  - 검색어 (query string)을 갖고 있는 웹페이지들 중에서 구함.
- Concentrate on links that go to other domains.
  - Local links have mainly navigational purposes. 따라서 같은 도메인내의 다른 페이지를 가리키는 링크는 제외함.
- Apply iterations of algorithm to reduced set of web pages.

HITS steps 1 - root and base set

HITS는 다음과 같은 조건을 만족하는 웹페이지들의 집합 (S_σ, base set) 을 대상으로 HITS 알고리즘을 적용한다.

(i) S_σ is relatively small. 즉, base set S_σ 내의 웹페이지들의 숫자가 많으면 곤란하다. HITS 계산이 장난 아니기에.

(ii) S_σ is rich in relevant pages. 즉, 물론 base set 내의 있는 페이지들이 검색어와 관련 깊으면 좋다.

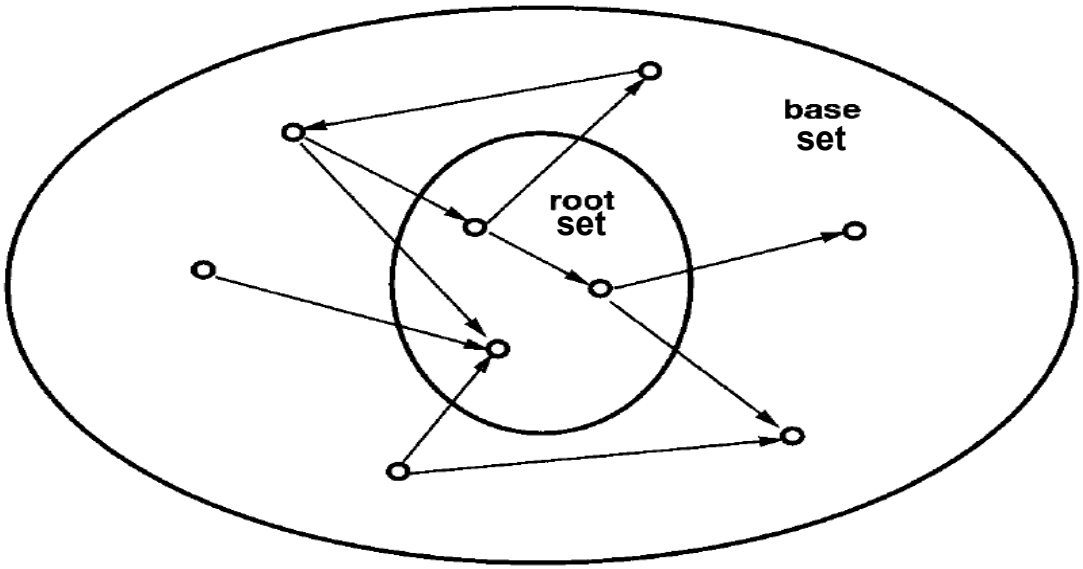
(iii) S_σ contains most (or many) of the strongest authorities. Authority 값이 높을 가능성이 많은 페이지일수록 좋다.

σ: query string, 검색어

Q_σ : Set of of all pages containing the query string σ, 검색어를 갖고 있는 페이지들. 검색엔진이 return 해주는 페이지들

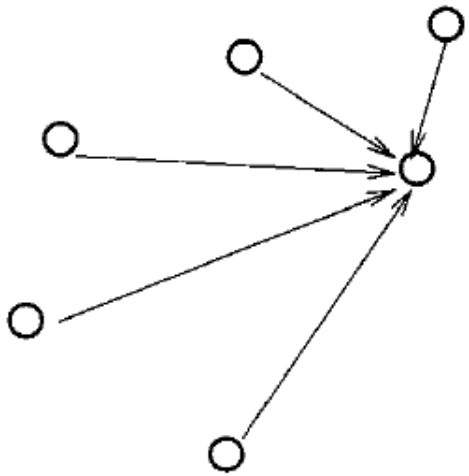
R_σ : Root set. t highest-ranked pages returned from the query of a text-based search. The links between the nodes in the R_σ is small. 즉 단순 text based 검색엔진에서 상위로 랭크된 페이지들인 R_σ 내의 페이지들간에는 서로 링크가 별로 없는 경우가 많다.

S_σ: Root Set 를 확장한 것. Root set 내의 페이지가 가리키는 페이지들과 와 Root set 내의 페이지를 가리키는 페이지들의 합집합. 단, Root set 내의 페이지를 가리키는 모든 페이지들을 포함하는 것이 아니라 일부만 포함시킨다.

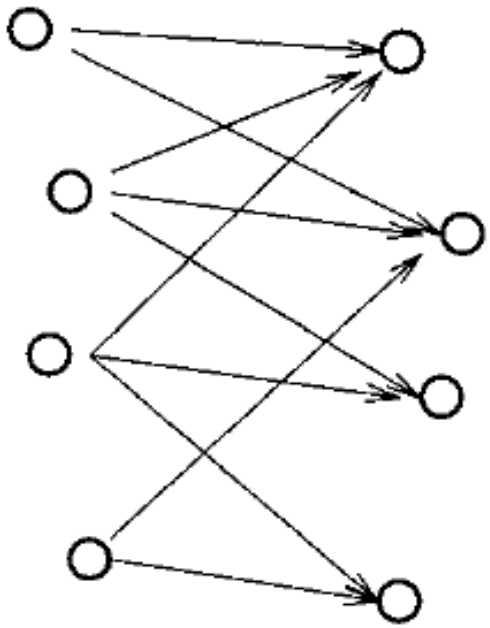


HITS steps 2 - what to count?

- Distinguish the pattern of relevant pages
  - 단순히 in-degree 만 감안하면 강한 authority 페이지와 단순히 인기가 많은 노드가 구별되지 않는다



unrelated page
of large in-degree



hubs authorities

✓ **good *hub* is a page that points to many good authorities; a good *authority* is a page that is pointed to by many good hubs.**

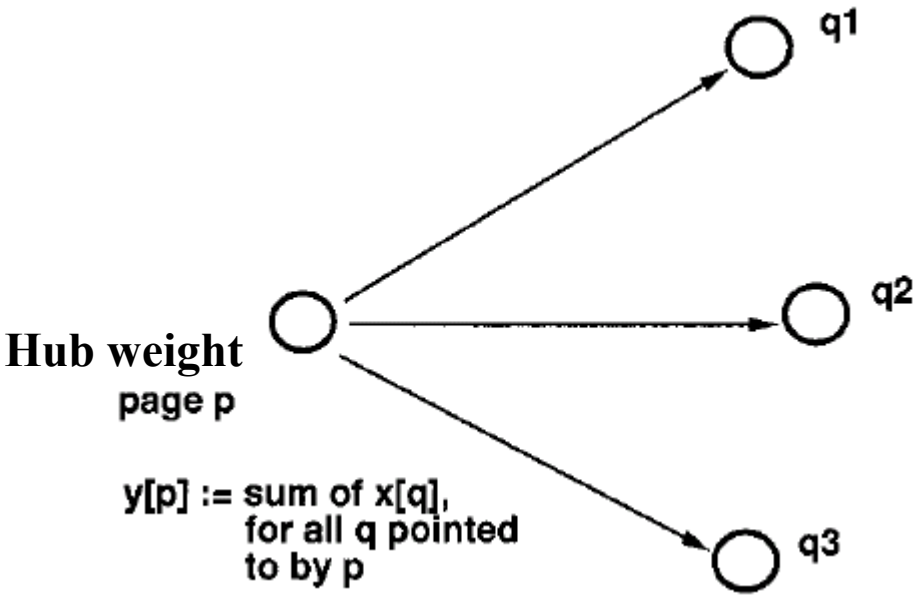
HITS - calculating weights

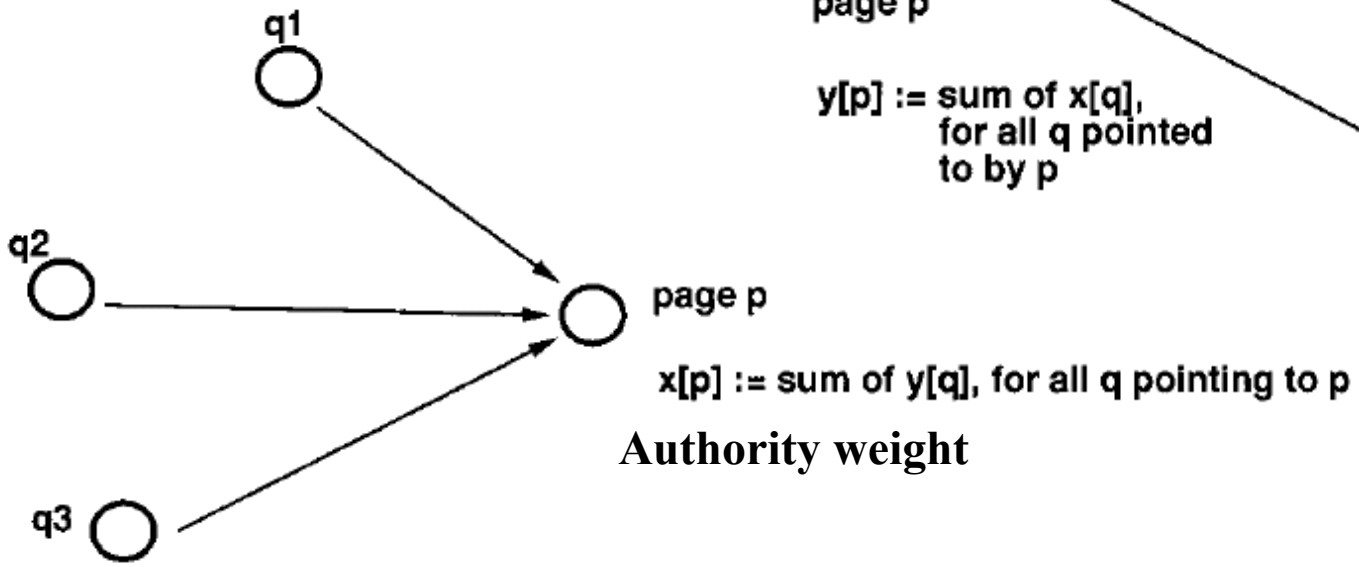
- Authority weight : $$x_p = \sum_{q \text{ such that } q \to p} y_q$$
- Hub weight : $$y_p = \sum_{q \text{ such that } p \to q} x_q$$

✓ **If *p* points to many pages with large *x*-values, then it should receive a large *y*-value; and if *p* is pointed to by many pages with large *y*-values, then it should receive a large *x*-value.**

HITS steps - how to count?

- Count OUT links
- Count IN links





✓ **Authority weight**를 구해야 **Hub weight**를 계산하고, **Hub weight**가 있어야 **Authority weight**를 계산할 수 있다. 이 사이클을 반복한다.

- Matrix notation: A - adjacency matrix
  - A[i, j] = 1 if i-th page points to j-th page
  - A[i, j] = 0 if i-페이지에서 j-페이지로의 링크가 없으면

$X = [x_1, x_2, , , , , x_n]^T$

$Y = [y_1, y_2, , , , , y_n]^T$ 로 하면

$X \leftarrow A^T Y$

$Y \leftarrow AY$

$$x \leftarrow A^T y \leftarrow A^T Ax = (A^T A)x$$

$$y \leftarrow Ax \leftarrow AA^T y = (AA^T)y$$

HITS - 결과

- Applying iterative multiplication (power iteration) will lead to calculating eigenvector of any "non-degenerate" initial vector.
- Hubs and auhtorities as outcome of process.
- Eigenvector/Hub & authority 값 --> not a process artifact. Hub나 authority 들의 초기값에 관계없이 순전히 link 구조에 따라 결정됨
- HITS 적용 후 높은 Hub 값과 Authority 값을 나타내는 페이지들의 링크 관계는 높은 Hub 값을 갖는 페이지들로부터 높은 Authority 값을 갖는 페이지들로 향하는 링크들이 빽빽이 있는 구조를 나타낸다.
- HITS에서 검색어가 사용되는 것은 root set을 구할 때까지이다. 이 이후로는 링크관계만 사용되고 검색어는 더 이상 Hub 값과 Authority 값을 구할 때 사용되지 않는다. 그렇지만 HITS 는 꽤 괜찮은 검색결과를 나타낸다. 예를 들어 "search engines" 로 검색했더니 Yahoo!, Excite, Magellan, Lycos, and AltaVista 가 높은 순위로 나왔다.

HITS 문제점

- From narrow topic, HITS tends to end in more general one. 세부적인 질의어를 던지면 보다 일반적인 결과를 가져올 경우가 있다.
- Specific of hub pages - many links can cause algorithm drift. They can point to authorities in different topics. 'A'란 주제에 대해 높은 허브값을 갖는 페이지_A가 다른 주제 B나 C에 관한 내용도 있어, 이 B, C 들이 해당 페이지_B와 페이지_C 들을 가리킬 때, 페이지_B와 페이지_C 에 'A'란 주제에 관련해 authority 값을 높여주게 된다. 이러면 사실 페이지_B와 페이지_C 들은 A 주제와 관계가 없는 데에도 A에 힘입어 authority 값이 높아지게 되어 문제가 된다.
- Pages from single domain / website can dominate result, if they point to one page - not necessary a good authority. 같은 도메인/웹사이트에 있는 페이지들이 어쩌다 base set에 많이 포함되고, 이 페이지들이 (같은 사람에게서 만들어 졌을 가능성이 많기에) 특정한 페이지를 많이 가리키면 별로 의미가 없는 그 페이지가 높은 authority 값을 갖게 된다.

HITS Algorithm의 개선

- Use weighted sums for link calculation.
- Take advantage of "anchor text" - text surrounding link itself. 링크의 anchor text는 자신이 가리키는 페이지의 특징/tag를 나타낼 경우가 많다. 즉, "자동차" 페이지를 가리키는 anchor text로 "요리"를 쓸 가능성이 많지 않다. 따라서, 검색어에 대해 authority가 높은 페이지들은 in-link의 anchor text가 검색어와 관련있는 단어일 경우가 많겠다. 이를 이용해 authority 계산에 in-link의 anchor text 의미가 유사하면 그 weight를 높인다.
- Break hubs into smaller pieces. Analyze each piece separately, instead of whole hub page as one. 앞에서 보았듯이 한 페이지에 많은 주제가 있을 시, 그 허브값이 분산되어 out-link 페이지들의 authority 계산에 쓰인다. 이럴 경우 하나의 페이지를 버츄얼하게 주제에 따라 분할한다.
- Disregard or minimize influence of links inside one domain.

Usage: trawling the Web

- 잘 보이지 않는 사이버 커뮤니티의 발견
  - group of content creators sharing a common interest with set of Web pages
  - large number of small communities with narrow topic
  - even not registered in newsgroups or other catalogs

How to find cyber communities

- Community - small group that has a dense pattern of linkage.
- Each community has similar linkage “signature”.
- Graph structure - directed bipartite graph

검색에서 Link 분석 음미

이 논문들은 1997-99 년경에 발표된 것이다. 따라서 이 즈음에 검색 품질을 높이는 방법으로 페이지내의 링크관계를 이용하려는 생각이 활발했음을 알 수 있다.

우리는 구글 검색엔진에서 페이지랭크를 보았고 또 비슷한 시대의 HITS를 보았다. 대학원생이던 Page와 Brin은 구글을 만들었고 교수이던 Kleinberg는 논문을 썼다. 어떤 사람들은 HITS가 페이지랭크에게 영감을 주었다고 한다.

페이지랭크값은 검색어와 관련없이 순수하게 링크연결이 의미하는 voting 지표에 따라 나온 값이지만 HITS의 HUB값과 AUTHORITY값은 검색어와 관계가 있다. 그러니, 같은 hub-authority 값을 지닌 페이지들이라도 주제가 다르면 절대적인 잣대로 중요성을 비교할 수 없을 것이다. 검색어가 다르지만 각각의 검색어내에서 hub-authority 값이 같은 페이지는 그 검색어내에서 상대적인 중요성 레벨이 같다고 생각할 수 있을 것 같다.

HITS에서 base set을 선정 시 우선 기본적으로 순수 text based search를 해서 그 중 가능성 있는 페이지들로 base set를 결정하는 데, 이 때 root set를 갖고 connected component 특성을 갖는 방향으로 성장시킨다. 앞서 우리는 A. Broder 일당들의 논문에서 웹의 그래프 구조를 보았다. 거기에서 웹의 약 ¼ 정도가 strongly connected component라고 보았다. 그러면, 검색어에 대해 순수 text based search 한 결과인 Q_σ 에 대해 가장 큰 **strongly connected component를 구해 여기에 HITS 를 적용하면 어떻게 될까?** 이 때에 strongly connected component 크기가 ¼ 이 넘을까? 너무 size 가 큰가? **HITS를 검색어와 상관없이 페이지랭크와 같이 전체웹에 적용하면 어떨까?**

한 페이지에 Hub값과 Authority값이 동시에 존재하는데, 이것을 갖고 랭킹을 한다면 이 값들을 어떻게 조합해서 쓰는 것이 좋을까? 검색어에 따라 달라질 수 있나? 컨텐츠 종류에 따라 달라질 수 있나?