지상(紙上) 블로그

불리안 모델과 벡터 모델
(Boolean and Vector Model)

심 경
정보학박사
한국도서관협회 평생회원
(주)아이리스닷넷 대표
shim@irisnet.co.kr

우리나라에 도서관자동화시스템이 보급된 지도 30여 년이 되었다. 속도와 기능이 빨라지고 다양해졌지만 변하지 않은 것은 이들 시스템의 기반이 되는 검색모델이다. 우리가 매일 사용하는 도서관자동화시스템에서 원하는 검색결과를 제공하는 역할을 하는 것을 검색 모델이라 하고, 그 가운데 가장 잘 알려져 있고 널리 사용되어 온 것은 불리안 모델(Boolean Model)이다. 불리안 모델이란 우리가 고등학교 시절 배운 불리안 대수(Boolean Algebra)와 집합이론(Set Theory)과 같은 수학에 근거한 검색 모델이다. 사실, 주의 깊게 보지 않았을 뿐 우리가 검색에 사용하는 모든 모델은 수학이론을 검색상황에 적용한 것이다. 예를 들면, 불리안 이외에도 벡터 모델, 확률 모델과 이들의 변형 또는 확장 모델이 존재하지만 결국 이들 모두는 수학이론을 기반으로 하고 있다.

불리안 모델이 말없이 우리 곁을 지켜주고 있었다면 가끔 웹이나 학술지 등에서 눈에 띄는 벡터 모델이라는 것이 있다. 우리가 어디서 검색결과가 유사도 순으로 정렬되었다고 하면 그 뒤에 있는 검색모델이 벡터 모델이 아닐까 의심해 볼 필요가 있다.[1)]

위에 언급한 것처럼 웹이나 학술지를 보면 가끔 데이터 마이닝(Data Mining) 또는 최신 정보 주지서비스(Selective Dissemination of Information: SDI)에서 벡터 모델을 사용하였다는 말을 들을 수 있다. 그런데 이 벡터 모델이라는 것이 새로운 기술일까? 아니다. 벡터 모델은 우리가 가장 자주 접하는 도서관자동화

1) 물론 일부 웹 포털에서는 벡터 모델에 의한 적합성 순위화보다 자주 검색되는 대상에 보다 높은 가중치를 주어 검색결과를 나열하기도 함.

시스템이나 웹 포털에 거의 사용되지 않아 친숙하지 않을 뿐, 학교시절 전공교재를 다시 열면 언제부터 거기 있었는지 모르지만 떡 하니 자리를 잡고 있다.

이 글에서는 벡터 모델이 무엇인지 그 원리를 알아보고 장단점을 알아보고자 한다. 특히 불리안 모델의 대체물 또는 보다 나은 모델인 것처럼 느껴지는 벡터 모델에 대한 간략한 이론적 설명과 더불어 그러한 이론적 배경이 검색결과에 미치는 영향을 살펴봄으로써 독자 스스로가 그 특징과 장단점을 유추할 수 있도록 시도하였다.

새로운 대상에 대한 접근법으로 우리가 이미 아는 것과의 차이점을 비교하면서 가지고 있는 지식에 새로운 지식을 쌓아가는 방법은 효과적이다. 따라서 먼저 불리안 모델의 특징을 간략히 살펴보고 그와 달리 벡터 모델에서 발생하는 검색과정의 상이성과 검색결과의 차이를 관찰해 보자.

불리안 검색 모델의 원리

불리안 모델에서 검색문(query 또는 need representation)은 용어의 불리안 로직으로 표현되고, 문헌 또는 문헌색인(index 또는 text/document representation)은 용어들의 집합으로 나타낸다. 한 문헌은 색인어의 집합으로 표현되므로 용어/색인어가 해당 문헌에 출현하면 "1"값을 가지며, 출현하지 않으면 "0"값을 가지게 된다.[2)]

불리안 검색에서 불리안 연산자를 사용하여 구성된 검색문이 검색하는 것은 도치색인 또는 도치파일(Inverted Index/file)이라 불리는 색인파일이다. 이 색인파일은 종종 포스팅 파일(postings file)이라고도 불리는데, 도치파일이라 칭하는 이유는 검색대상이 되는 문자나 숫자와 같은 콘텐츠를 그들이 출현하는 위치로 매핑(mapping)해 주기 때문이다. 이와 같이 도치색인에서 자모순으로 배열되는 "문자나 숫자"를 우리는 색인 엔트리(index entry)라고도 한다.

복잡한 설명보다 실제 예를 들어서 살펴보자. 〈그림 1〉은 불리안 검색문과 문헌색인의 예이다.

2) 물론 불리안 모델에 가중치를 용어에 부여하지 않으며, 부여하는 것이 가능하지도 않음. 이 표현의 의미는 불리안 모델에서 문헌은 용어집합이므로 그 용어가 가지는 값은 "1"이거나 "0", 즉 그 집합의 멤버인가 아닌가 만이 존재한다는 것임. 따라서 해당 데이터베이스에 존재하지만 특정 문헌의 색인에 사용되지 않은 색인어의 값이 0인 것이며, 색인되지 않은 용어를 우리는 굳이 표현하지 않음. 예를 들면, Di={검색, 이론, 정보, 텍스트}로 표현하지, Di={검색(1), 시맨틱 웹(0), 온톨로지(0), 이론(1), 정보(1), 텍스트(1) …}처럼 해당 문헌에 색인어로 채택되지 않은 용어까지 값이 0임을 밝히면서 표현해 주지 않는다는 의미임.

검색문 = ((text OR information) AND retrieval AND NOT theory)

$문헌_{\#1}$ = (information, retrieval, theory)

$문헌_{\#2}$ = (text, retrieval)

$문헌_{\#3}$ = (information theory)

.
.
.

〈그림 1〉 검색문과 문헌색인 예시

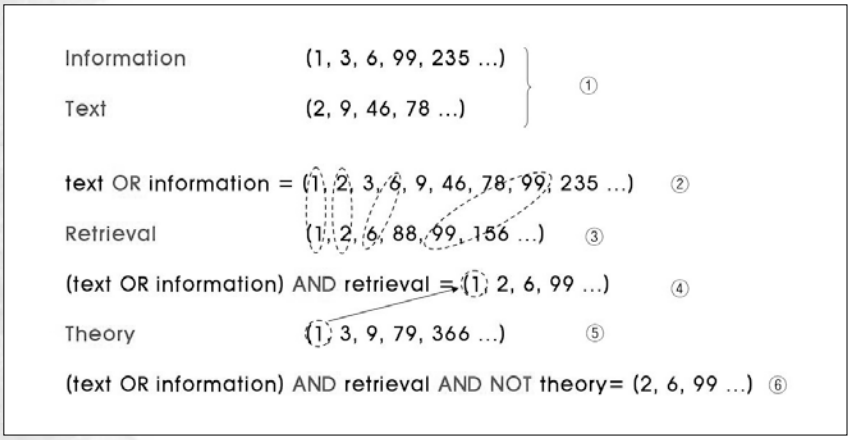
이 검색문은 text retrieval이나 information retrieval에 관한 것이나, theory를 포함하지 않은 자료를 이용자가 찾고자 하는 것을 알 수 있다. 그 아래 문헌 리스트는 개별 문헌이 색인된 색인어들을 색인어의 집합으로 보여주고 있다. 앞서 간략히 설명한 바와 같이 검색시스템이 위 검색문을 전달받으면 위에 예시한 모든 문헌을 일일이 뒤지며 해당 문헌에 포함된 색인어를 확인하는 것이 아니라 도치색인을 먼저 검색한다. 이와 같이 색인을 먼저 검색하는 절차는 우리가 정보검색 관련 문헌에서 "정확률"이라는 용어를 찾을 때, 책의 맨 앞에서 마지막 장까지 읽지 않고, 책 말미에 마련된 색인을 먼저 찾아 해당 페이지로 이동하는 이치와 동일하다.

색인 엔트리	문헌 번호
Communication	(21, 33, 35, 56 ...)
Information	(1, 3, 6, 99, 235 ...)
Retrieval	(1, 2, 6, 88, 99, 156 ...)
Text	(2, 9, 46, 78 ...)
Theory	(1, 3, 9, 79, 366 ...)

.
.
.

〈그림 2〉 도치파일의 예

〈그림 2〉의 도치색인은 색인어를 엔트리로 삼아 이들의 자모순으로 배열된다. 따라서 위 검색문은 먼저 text와 information을 색인(〈그림 3〉의 ①)에서 찾고 이들을 포함한 문헌번호의 합집합(〈그림 3〉의 ②)을 구한다.



〈그림 3〉 불리안 검색과정

다음 과정은 이들 문헌집합 중 retrieval이란 용어를 포함한 문헌(〈그림 3〉의 ③)만을 추출해야 하므로 이 조건을 만족하는 교집합을 구하면 〈그림 3〉의 ④가 된다. 이와 같이 불리안 모델에서는 검색문과 문헌색인의 비교 시, 검색문에 표현된 용어 및 구문과 정확히 일치하는 문헌만을 검색할 뿐, "가장 유사하거나 가까운(closest 또는 best match) 문헌"을 인정하지 않는다.[3] 따라서 이런 검색기법을 완전매칭(Exact Matching)기법이라고 한다.

불리안 검색 모델의 장단점

검색 모델의 장단점이란 상대적인 것이지만, 불리안 모델의 장점은 검색문의 논리적 구조와 이 모델이 사용하는 도치파일을 통한 검색 효율성을 든다. 첫 번째의 논리적 구조는 불리안 연산자로 용어 간의 관계설정이 용이한 유일한 검색 모델이라는 점이며, 둘째의 검색효율성이란 앞서 설명한 도치색인에 대한 이진검색(binary search algorithm 또는 binary chop)으로 검색속도가 현저하게 빠른 것이다[4]. 반면, 단점으로는, 첫째, 검색문과 정확히 일치하지는 않으나 적합한 문헌을 검색하지 못하며, 둘째, 검색결과의 순위화가 불가하고, 셋째, 용어의 상대적 중요성을 나타내는 가중치를 검색문이나 색인에 반영할 수 없으며, 넷째, 검색문에 요구되는 복잡한 불리안 로직과, 다섯째, 검색문과 색인의 용어가 동일한 시소러스에서 추

3) 다시 말하여, 위 검색문에 포함된 text, information, retrieval 중 2개의 용어가 일치하므로 비슷한 문헌이라고 검색하지 않음.
4) 이진검색이란 위 검색문의 retrieval을 가진 문헌을 찾기 위하여 색인 순차검색(sequential search)을 하는 것이 아니라 자모순으로 배열된 색인의 중간, 즉, "M"과 retrieval의 "R"중 어느 것이 큰 값을 가지는가를 비교함. M < R이므로 시스템은 A부터 M까지는 고려하지 않고, 나머지 자모의 중간인 "S"와 "R"의 값을 비교 후, S > R이므로, 다시 S에서 Z까지는 검색대상에서 제외하는 식으로 진행함으로써 검색 대상을 절반씩으로 줄여나가는 방법임.

출된 것을 전제로 한다는 것이다.[5]

타 검색모델에는 없는 검색문 구조 때문에 불리안 모델을 발전시키거나 다른 모델에 합쳐보려는 시도가 여러 차례 있었다. 예를 들면, 확장불리안 모델(Extended Boolean Model)[6]과 확률검색에 불리안 검색문을 활용하는 방법[7] 등 불리안 검색문 구조를 유지하면서 검색결과 순위화의 특징을 포함하려는 노력이다.

그런데 왜 불리안 모델이 여전히 널리 보급되어 있는 것일까? 그 답은 의외로 간단하다. 불리안 모델이 다른 검색 모델에 비하여 먼저 보급되었으며, 무엇보다도 새로운 검색 모델이 불리안 검색을 기반으로 한 기존 시스템들을 모두 대치할 만큼의 성능향상을 보이지는 못하기 때문이다.[8]

벡터 모델의 역사

벡터 검색 모델은 영어로는 Vector Model, Vector Space Model 또는 Term Vector Model 이라고 한다. 이 모델이 처음 소개된 것은 1960년대에 시험용 검색시스템인 SMART(System for the Mechanized Analysis and Retrieval of Text)를 사용하여 각종 검색시험을 수행한 SMART 프로젝트이다.[9] 벡터 모델은 정보검색환경에 벡터라는 수학이론을 적용한 것으로 앞서 언급한 불리안 모델의 단점을 보완한 모델이라고 할 수 있다. 이는 검색되는 문헌이 반드시 검색문의 용어와 일치하지 않아도 둘 사이의 유사도를 계산하여 가장 유사한 문헌부터(closest 또는 best match) 순위화하여 검색하므로 부분매칭(Partial Matching)기법이라고 한다. 그럼 어떻게 검색을 하길래 이런 것이 가능할까? 그 답은 벡터라는 개념과 그 속성에 달려있다.

벡터란 무엇인가?

벡터란 기초수학, 물리나 공학에서 방향성과 길이(크기)를 가진 객체를 말한다. 정보검색에 벡터공간모

5) Belkin, N.J. & Croft, W.B. (1987). Retrieval techniques. In M.E. Williams (Ed.). *Annual Review of Information Science and Technology*, 22, 109~145.

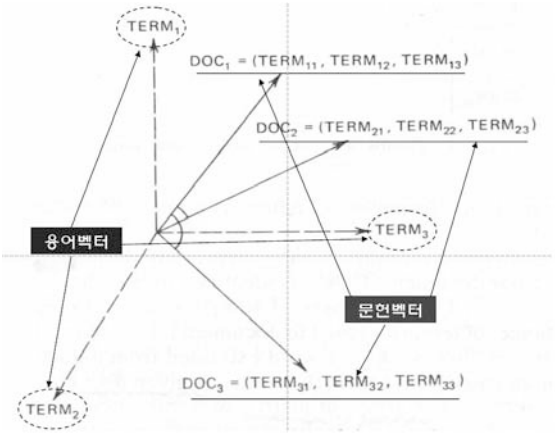
6) Salton, G., Fox, A. & Wu, H. (1983). Extended Boolean information retrieval. *Communication of the ACM*, 26, 1022~1036.

7) Radeki, T. (1982). A probabilistic approach to information retrieval in systems with Boolean search request formulations. *JASIS*, 33, 365-370; Croft, W.B.(1986). Boolean queries and term dependencies in probabilistic retrieval models. *JASIS*, 37, 71-77.

8) 과거 수십, 수백 개의 정보검색성능연구를 살펴보아도 불리안 모델과 비교하여 정확률과 재현율을 현격히 향상시킨 모델은 없었음. 이와 같은 조건에서 약간의 성능이 우수한 검색모델로 교체하려면 추가비용이 투입되어야 하고 이는 비용 대비 효과 측면에서 정당화가 어려움.

9) SMART 프로젝트의 공식 보고서 격인 Salton, G. (Ed.) (1971). *The SMART retrieval system: experiments in automatic document processing*. Englewood Cliffs: Prentice-Hall을 참조하면 상세 내용을 알 수 있으며, 이 프로젝트에서는 다양한 검색모델, 적합성 피드백, 이들의 성능평가, 성능척도 등 정보검색에 대한 폭넓은 연구가 수행되었음.

델을 적용한다는 것은 개별용어, 그리고 용어들의 집합이라 볼 수 있는 검색문과 문헌색인을 벡터공간에 표현할 수 있다는 가설을 기반으로 한다. 이 말의 의미는 이론적으로 n-차원 공간(n-dimensional space)에 벡터를 표현할 수 있다는 것이며, 〈그림 4〉와 같이 용어를 축으로 하면 용어 수만큼의 차원이 생기고 그 공간에 여러 용어의 집합인 검색문이나 문헌색인을 벡터로 표현할 수 있다는 것이다.



〈그림 4〉 문헌의 벡터공간 표현 [10]

이렇게 용어를 벡터로 표현하는 것은 개별 용어를 하나의 차원으로 보고 각 용어가 가진 가중치를 벡터의 길이로 나타냄으로써 가능하다. 그렇다면 여러 용어의 집합인 검색문이나 문헌색인은 어떻게 표현해 주는가가 궁금해 진다. 그런데 교재에 나오는 표현은 항상 아래와 같다.

$$D_i = (a_{i1}, a_{i2}, \ldots, a_{it})$$
$$Q_j = (q_{j1}, q_{j2}, \ldots, q_{jt})$$

〈그림 5〉 문헌과 검색문의 벡터표현

위에 보인 문헌과 검색문은 일단 영어문자 아래 첨자가 두 개씩 붙어있어 처음부터 이해하고자 하는 의지를 꺾어놓기 일수지만 조금만 인내심을 갖고 보도록 하자. 잠깐, 이들에 대한 상세한 설명을 하기 전에 벡터 모델에서 검색문은 불리안 연산자없이 단순히 용어를 나열하고 있음을 주지할 필요가 있다. 위

10) Salton, G. & McGill, M.J. (1983). *Introduction to modern retrieval*. New York: McGraw-Hill. p. 122.

표현에서 먼저 D는 문헌색인을, Q는 검색문을 의미한다. 그런데 아래 첨자 중 첫 번째 것은 문헌색인과 검색문이 여러 개 존재할 수 있다는 의미인 것을 알겠는데, 공교롭게도 1에서 t까지로 모두 일치하는 두 번째 첨자는 뭘 의미하는 것일까? 짧게 말하여 이는 벡터 모델에서 사용하는 방법으로 해당 시스템에서 사용되는 전체 용어의 수를 의미하며, ai1 또는 qj2는 개별 용어의 벡터값을 나타낸다. 이 개념은 〈그림 6〉을 보면 쉽게 이해할 수 있다.

용어 / 문헌/ 질의문	1	2	3	4	5	6	7	…	t
예	검색	문헌	온톨로지	정보	컴퓨터	탐색	텍스트		퍼지모델
문헌 1	1	1	0	1	0	0	1		0
문헌 2	1	0	1	1	0	0	0		0
문헌 3	0.8	0	0.5	0.9	0	0	0		0
문헌 4	0.5	0.3	0.9	0.7	0	0	0		0
검색문 1	1	0	0	1	0	0	0		0
검색문 2	0.9	0	0	0.8	0	0	0		0

〈그림 6〉 문헌 및 검색문의 벡터표현

〈그림 6〉의 맨 윗줄의 숫자는 〈그림 5〉의 1부터 t까지를 의미하며, 곧 t는 해당 데이터베이스에서 사용된 용어의 총수가 된다. 그 아래 용어는 숫자로 표시된 용어들을 예로 넣은 것이다. 따라서 〈그림 5〉의 Di는 문헌1에서 문헌4까지를 의미하며, 문헌 1과 2는 가중치가 부여되지 않은 경우이며, 문헌 3과 4는 가중치가 부여된 것이다. 여기서 0값을 가진 용어는 색인어로 부여되지 않은 용어를 의미한다. 만약 〈그림 5〉의 두 번째 아래첨자에 대한 이해하기 어려웠다면 이는 바로 0값을 가진 용어도 표현해 주었기 때문이다.

벡터 모델에서 검색은?

벡터 모델에서 위와 같이 검색문과 문헌색인을 표현해 준다면 이들 간의 검색은 어떻게 할까? 개념적으로는 〈그림 4〉에 표시한 벡터공간에서 두 벡터의 각도가 적으면 유사하다고 판단하나 이는 이론적 또는 개념적으로 그렇다는 것이지 실제로 벡터공간에서 두 벡터 간의 각도를 측정하거나 벡터의 크기까지를 고려하는 것은 불가능하다. 따라서 두 벡터 간의 유사도는 계산으로 산출되며, 이를 가능하도록 하는 것은 실수를 대상으로 더하기, 빼기, 곱하기와 부정과 같은 기본적 대수원칙을 준수하는 벡터의 속성 때문이다.

두 벡터의 유사도를 계산하는 방식은 가장 간단한 내적(Inner Product), 코사인 계수(Cosine coefficient), 자카드 계수(Jaccard coefficient), 다이스 계수(Dice coefficient)등 다양한 방법이 있으며[11], 이들에 따른 검색효과의 차이는 크지 않은 것으로 알려진다. 이들 유사도 계수는 〈그림 6〉에 제시된 용어에 가중치 부여 여부와는 무관하게 검색문과 문헌의 유사도를 정도(degree)로 제시하여 검색결과의 적합성 순위화가 가능하도록 해 준다. 즉, 〈그림 6〉의 문헌1에서 문헌4까지 모두 순위화 된 결과를 제시할 수 있다는 의미이다.

벡터 모델의 장단점

벡터 모델의 장점은 첫째, 검색문이나 문헌색인 용어의 가중치 부여에 관계없이 유사도가 계산되므로 검색결과를 적합성 정도에 따라 순위화가 가능한 것이며, 둘째, 검색문에 복잡한 불리안 연산자를 사용하지 않는다는 것이다. 사실 이 두 번째 장점은 영어권 사람들에게 해당되는 것이며 우리처럼 비영어권에는 불리안 연산자가 그리 어렵지 않다. 영어권에서 불리안 연산자를 어려워하는 이유는 불리안의 AND나 OR이 일반 영어에서 의미하는 바와는 거의 반대 의미로 해석되기 때문일 것이다.

비록 벡터 모델이 불리안 모델에 결핍된 검색결과의 순위화라는 대업(?)을 달성하였지만, 이 모델은 치명적인 약점을 가진다. 벡터 모델에서 검색을 위하여는 검색문 벡터와 모든 문헌벡터를 순차적으로 하나씩 비교를 하여야 하는데, 이는 문헌집단의 규모가 커지면 거의 검색속도를 예측할 수 없을 만큼 시스템 반응속도가 느려질 수 있는 것이다. 그러므로 벡터검색은 인터넷과 같이 대용량 데이터 검색에 대처할 수 없다는 태생적 한계점이 있다고 할 수 있다.

이러한 벡터 모델의 약점을 극복해 보려는 노력이 없었던 것은 아니다. 예를 들면, 문헌집단 안에 있는 비슷한 문헌들을 그룹으로 묶어서 검색문과 비교할 시 전체 문헌이 아닌 미리 나누어 놓은 그룹들(클러스터; clusters)의 표현(representations)만 비교하여 가장 유사한 그룹을 검색하는 클러스터 모델(Cluster Model)이 하나이며, 벡터파일 이외에 도치색인(Inverted File)을 활용하여 최소한 하나의 색인어라도 검색문의 검색어와 일치할 경우 그 문헌들에 대하여만 유사도를 계산하는 방법이 시도된 적이 있다[12].

---

11) 이들의 종류, 속성과 수학적 설명 등에 관심이 있는 독자는 Salton, G. & McGill, M.J. (1983). *Introduction to modern retrieval*. New York: McGraw-Hill. pp. 201-204; van Rijsbergen, C.J. (1979). *Information retrieval*. 2nd ed. London: ButterWorths. pp. 38-42를 참조할 것

12) Smeaton, A.F. & van Rijsbergen, C.J. (1981). The nearest neighbor problem in information retrieval: an algorithm using upperbounds. In C.J. Couch (Ed.) *Proceedings of the 4th International Conference on Information storage and Retrieval: theoretical issues in information retrieval* (pp. 83~87). New York: ACM; Murtagh, F. (1982). A very fast exact nearest neighbor algorithm for use in information retrieval. *Information Technology: research and development*, 1, 275~283.

결 언

정보검색연구에서 수많은 검색모델 간의 성능시험결과가 보고되었지만 특정모델이 다른 모델보다 월등한 성능을 보인 것은 없다. 그러나 이러한 시험을 통하여 배운 것은 유사한 검색성능에도 불구하고 이들이 검색한 결과세트에 포함된 적합문헌이 서로 다르다는 것이다[13]. 또한 이러한 선행연구결과를 기반으로 모든 검색모델과 정보요구를 표현하는 방법을 통합하는 모델도 제시되었으나 역시 그 검색효과가 놀랄 만큼 향상을 보이지는 못했다.[14]

이 글에서 살펴본 불리안 모델과 벡터 모델을 비교할 때, 서로의 상대적 장단점을 가진다. 그러나 실무적 측면에서 벡터 모델은 대규모 데이터베이스에 적용되기에는 한계가 있는 것으로 판단된다. 따라서 벡터 모델을 실무에 적용하려면 가능한 대규모 순차비교가 발생하지 않는 특수한 상황을 찾아야 할 것이다. 필자의 견해로는 도서관 현장에서 불리안 모델을 배제하고 벡터 모델을 적용할 상황을 쉽게 찾아보기 힘들다. 설사 발견한다 할지라도 성능향상이 없는 복잡한 알고리즘으로 시스템 반응시간을 초조하게 기다릴 만큼 벡터 모델을 채택할 이론적 또는 실무적 이유를 찾기는 어렵다.

"새 술은 새 부대에 담아야 한다(new wine, new wineskins)"는 말이 있지만, "구관이 명관이다(舊官名官)"라는 말도 있다. 글쎄, 벡터 모델에 관한 한 도서관환경에서 아직까지는 후자가 더 맞는 말인 것 같다.

알 림

〈〈도서관문화〉〉 2008년 1월호를 시작으로 '지상(紙上) 블로그' 코너를 집필해오시던 심경 박사의 글은 이번호로 끝마칩니다.
그동안 좋은 글을 써주신 필자께 감사드립니다.

13) Belkin, N.J. & Croft, W.B. (1987). Retrieval techniques. In M.E. Williams (Ed.). *Annual Review of Information Science and Technology*, 22, 109-145.

14) Turtle, H. & Corft, W.B. (1990). Inference Networks for Document Retrieval. *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, 1-24. New York: ACM Press; Turtle, H. & Croft, W.B. .(1991). Evaluation of an inference network-based retrieval model. *ACM Transaction on Information Systems*, 9(3), 187-222.