확장 불리언 질의에 대한 비용 기반 최적화*

Cost-based Optimization of Extended Boolean Queries

박병권(Byung-Kwon Park)**

초 록

본 논문에서는 역색인 파일을 이용하여 확장 불리언 질의를 처리할 때 최소 비용의 질의 처리 방법을 구해 주는 질의 최적화 알고리즘을 제시한다. 확장 불리언 질의를 처리하는 방법은 질의를 구성하는 키워드의 처리 순서에 따라 여러 가지가 있을 수 있으므로 확장 불리언 질의 최적화 문제는 결국 최적 키워드 처리 순서를 구하는 문제로 귀결된다. 본 논문에서는 이 문제가 데이터베이스 질의 최적화에서 최적 조인 순서를 구하는 문제와 구조적으로 유사함을 보이고 이 분야의 연구 결과를 이용하여 문제를 해결한다. 즉, 확장 불리언 질의 처리에 대한 비용 모델을 수립하고 키워드 선택률과 역색인 파일 접근 비용을 이용하여 키워드 순위 개념을 도입한 후 이를 이용하여 최적 키워드 처리 순서를 구하는 알고리즘을 도출한다. 그리고 도출한 질의 최적화 알고리즘의 최적성을 증명하고 실험을 통하여 실제로 최소 비용의 질의 처리 방법을 구함을 보이고, 질의 최적화를 하지 않을 경우와 비교하였을 때 그 성능이 월등히 우수함을 보인다. 본 논문에서 제시한 질의 최적화 알고리즘은 정보검색시스템의 질의 처리 성능 향상에 큰 기여를 하리라 믿는다.

ABSTRACT

In this paper, we suggest a query optimization algorithm to select the optimal processing method of an extended boolean query on inverted files. There can be a lot of methods for processing an extended boolean query according to the processing sequence of the keywords contained in the query. In this sense, the problem of optimizing an extended boolean query is essentially that of optimizing the keyword sequence in the query. In this paper, we show that the problem is basically analogous to the problem of finding the optimal join order in database query optimization, and apply the ideas in the area to the problem solving. We establish the cost model for processing an extended boolean query, and develop an algorithm to find the optimal keyword-processing sequence based on the concept of keyword rank using the keyword selectivity and the access costs of inverted file. We prove that the method selected by the optimization algorithm is really optimum, and show, through experiments, that the optimal method is superior to the others in performance. We believe that the suggested optimization algorithm will contribute to the significant enhancement of the information retrieval performance.

키워드: 정보검색, 확장 불리언 질의, 질의 최적화, 비용모델, 키워드 선택률, 비용지수, 순위,
information retrieval, extended boolean query, optimization algorithm

* 이 논문은 2000학년도 동아대학교 학술연구조성비(신진과제)에 의하여 연구되었음

** 동아대학교 경영정보과학부 전임강사(bpark@daunet.donga.ac.kr)

■ 논문 접수일 : 2001년 6월 26일
■ 게재 확정일 : 2001년 9월 17일

1 서 론

정보검색을 위한 질의에는 불리언 질의(boolean queries), 확장 불리언 질의(extended boolean queries), 벡터 질의(vector queries), 확률적 질의(probabilistic queries) 등이 있으나(Jones 1997; Korfhage 1997), 이 중 가장 기본적이고 대부분의 정보검색시스템이 지원하는 질의가 확장 불리언 질의이다(Korfhage 1997; Witten 1994). 확장 불리언 질의는 가중치(weight) 개념을 도입하여 불리언 질의를 확장한 것이다. 가중치는 키워드와 문서의 쌍에 대해 부여되는 것으로서 특정 문서에 대한 특정 키워드의 중요성을 나타낸다. 또한 질의 결과에 대해서도 각 문서마다 가중치가 부여되는데 이 가중치는 질의에 대한 그 문서의 적합한 정도를 나타낸다.

확장 불리언 질의를 처리하기 위해서는 질의에 포함된 각 키워드가 어느 문서에 포함되어 있고 가중치가 얼마인 지를 알려주는 색인 파일이 필요한데 이를 위해 많은 정보검색시스템이 역색인 파일(inverted file)을 사용하고 있다(Faloutsos 1985; Zobel 1998). 역색인 파일에는 각 키워드마다 포스팅 리스트(posting list)가 있고, 각 포스팅(posting)에는 그 키워드가 포함된 문서의 식별자와 그 문서의 가중치가 포함되어 있다(Salton 1988). 그리하여 역색인 파일은 확장 불리언 질의를 처리할 때 질의에 포함된 각 키워드에 대한 포스팅 리스트를 제공하는 역할을 한다.

역색인 파일을 통하여 확장 불리언 질의를 처리하기 위해서는 질의에 포함된 각 키워드에 대하여 포스팅 리스트를 구하고 이들을 확장 불리언 연산에 따라 병합(merge)하여야 한다. 그런데 문서의 추가, 삭제, 변경이 없는 정적 문서 집합의 경우에는 역색인 파일이 한번 만들어지면 변경이 일어나지 않으므로 포스팅 리스트들의 병합 연산을 빨리 하기 위해 역색인 파일을 만들 때 포스팅들을 문서식별자 순으로 정렬(sort)해 둘 수 있다. 그러나, 문서의 추가, 삭제, 변경이 빈번한 동적 문서 집합의 경우에는 역색인 파일이 만들어진 후에도 변경이 자주 일어난다. 특히, 문서가 추가됨에 따라 포스팅 리스트에 새로운 포스팅을 추가하려고 할 때 문서식별자 순의 정렬을 유지하려면 매번 포스팅 리스트에서 새로운 포스팅의 삽입 위치를 찾아서 삽입하여야 하므로 역색인 파일의 변경 비용이 크다.

따라서 새로운 포스팅을 추가할 때 항상 포스팅 리스트의 끝에 추가(append)하게 되면 포스팅 리스트는 문서식별자 순의 정렬을 유지하지 못하지만 포스팅의 추가가 간단해져 역색인 파일의 변경 비용이 작게 든다. 본 논문에서는 이러한 역색인 파일에 대한 확장 불리언 질의 최적화를 다룬다. 먼저 확장 불리언 질의에 대한 처리 방법이 여러 가지가 있을 수 있음을 보이고 그 중 비용이 최소가 되는 질의 처리 방법을 선정해 주는 질의 최적화 알고리즘을 제안한다. 또한 질의 최적화의 중요성을 평가해 보기 위해 질의 최적화를 수행할 경우와 수행하지 않을 경우의 질의 처리비용을 실험을 통하여 비교해 본다.

Kaszkiel(1998)은 확장 불리언 질의 처리 방식을 키워드 중심(term-oriented) 처리 방식과 문서 중심(document-oriented) 처리 방식으로 분류하였는데 본 논문에서는 문서 중

심 처리 방식에서의 질의 최적화를 다룬다. 문서 중심 처리 방식에서는 각 문서에 대해 차례로 주어진 질의를 만족하는 지를 평가하는데 이 때 질의 최적화 알고리즘은 어떤 키워드 순으로 평가할 것인가를 결정해 준다. 유사한 연구로서 Hanani(1977)는 데이터 레코드 집합에 대해 각 데이터 레코드가 불리언 식을 만족하는 지를 평가 할 때 불리언 식의 어떤 항(element) 순서로 평가하는 것이 최적인가를 연구하였다. 그리고 데이터베이스 분야에서는 Ibaraki(1984)가 데이터베이스 질의를 처리할 때 순위(rank) 개념에 기반하여 최적 조인 순서를 구하는 방법을 제시하였다. 이 순위 개념은 사용자 정의 함수를 가진 질의 처리에도 이용되고 있다(Chaudhuri 1999).

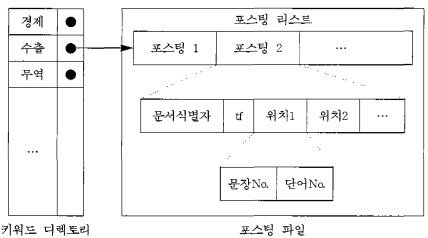
본 논문에서는 Hanani(1977)와 Ibaraki(1984)의 연구에서 제시된 질의 최적화 개념을 역색인 파일을 이용한 불리언 질의 최적화에 적용하였다. 이를 위하여 질의 처리 비용 모델을 수립하고 이 비용 모델이 Monma(1979)가 제시한 ASI(Adjacent Sequence Interchange) 성질을 만족하고 따라서 순위 개념에 기반하여 최적화할 수 있음을 보인다. 역색인 파일을 이용한 불리언 질의 최적화 연구로는 Liu(1976)가 주어진 불리언 질의와 동등한 질의 형태 중 포스팅 리스트의 병합을 가장 효율적으로 만드는 질의 형태를 구하는 최적화 알고리즘을 제시하였다. 본 논문에서는 Liu(1976)가 제시한 알고리즘을 통해 결정된 질의 형태를 가정하고 그 질의 형태에서의 최적 키워드 처리 순서를 다룬다. Moffat(1996)은 압축된 포스팅 리스트에 대한 불리언 질의 처리 속도를 높이기 위해 포스팅 리스트 내부에 색인 정보를 심는 방안을 제안하였고, Tomasic(1993)은 불리언 질의의 분산 처리 속도를 높일 수 있는 역색인 파일 저장 구조를 연구하였다. 그러나 이러한 연구들은 모두 비용에 기반한 체계적인 최적화는 제공하지 못하고 있다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 역색인 파일을 이용한 확장 불리언 질의 처리 방법을 살펴보고, 제 3 장에서는 확장 불리언 질의에 대한 질의 최적화 알고리즘을 제시하며 이의 최적성을 증명한다. 제 4 장에서는 질의 최적화 알고리즘의 성능을 실험적으로 평가하고 제 5 장에서 결론을 맺는다.

2 역색인 파일을 이용한 확장 불리언 질의 처리 방법

역색인은 키워드가 주어졌을 때 그것과 연관된 문서들의 식별자와 가중치를 찾을 수 있도록 각 키워드 별로 포스팅 리스트를 유지하고 있다. 〈그림 1〉은 역색인 파일의 한 예이다.

〈그림 1〉에서 키워드 디렉토리는 각 키워드마다 그 키워드에 대한 포스팅 리스트를 가리키는 포인터를 가지고 있고 포스팅 파일은 모든 키워드의 포스팅 리스트들을 가지고 있다. 따라서 키워드 디렉토리를 통해 특정 키워드의 포스팅 리스트를 찾을 수 있다. 포스팅 리스트에는 포스팅들이 차례로 들어 있고, 각 포스팅에는 그 키워드가 발생한 문서의 문서식별자, 키워드 발생회수(term frequency: tf),



〈그림 1〉 역색인 파일의 예

키워드 발생위치 등이 들어 있다. 키워드 발생위치는 키워드가 문서의 몇 번째 문장에서 발생했는가(문장No.)와 그 문장 내의 몇 번째 단어에서 발생했는가(단어No.)로 구성된다.

역색인 파일을 이용하여 확장 불리언 연산자를 처리하는 방법은 여러 가지가 있고 (Salton 1988; Witten 1994) Lee등(1993)은 검색 효율(retrieval efficiency)과 검색 효과(retrieval effectiveness)를 높일 수 있는 새로운 연산 방법을 제안하고 있다. 그러나 가장 간단한 연산 방법은 다음과 같은 퍼지 집합 모형(fuzzy set model)이다. 이 모형에서 연산자 AND는 다음과 같이 처리된다. AND로 연결된 두 키워드의 포스팅 리스트를 키워드 디렉토리를 통해 각각 구한 후에 왼쪽 키워드의 포스팅 리스트를 순차검색하면서 각 포스팅에 포함된 문서식별자와 가중치를 구하고 그 문서식별자가 오른쪽 키워드의 포스팅 리스트에 포함되어 있는 지를 탐색한다. 탐색이 성공하면 왼쪽 키워드의 가중치 값과 오른쪽 키워드의 가중치 값을 비교하여 작은 값을 문서식별자와 함께 질의 결과에 포함시킨다. 연산자 OR과 NOT의 처리도 AND와 동일하다. 다만 OR의 경우 가중치를 비교하여 큰 값을 질의 결과에 포함시키고, NOT의 경우 왼쪽 키워드의 가중치에서 오른쪽 키워드의 가중치를 뺀 값이 0 보다 클 경우에만 질의 결과에 포함시킨다.

3 질의 최적화 알고리즘

확장 불리언 질의가 n개의 키워드로 이루어져 있으면 질의를 처리하는 방법의 수는 키워드를 어떤 순서로 처리하느냐에 따라 $n!$개가 존재한다. 본 장에서는 $n!$개의 질의 처리 방법들 중에서 최소 비용을 가지는 질의 처리 방법을 찾기 위하여 먼저 확장 불리언 질의 처리에 대한 비용모델을 수립하고 이에 기반한 질의 최적화 알고리즘을 제시한다.

〈표 1〉 비용 파라미터

기 호	이 름	정 의
n_D	총 문서 개수	저장된 문서의 총 개수
$df(W)$	키워드 W의 문서 빈도수	저장된 문서들 중에서 키워드 W가 포함된 문서의 개수
$ps(W)$	키워드 W의 포스팅 리스트 순차 검색 비용	키워드 W의 포스팅 리스트 전체를 처음부터 끝까지 순차적으로 읽기 위해 필요한 디스크 페이지 입출력 횟수
$pa(W)$	키워드 W의 포스팅 리스트 탐색 비용	키워드 W의 의 포스팅 리스트에서 특정 포스팅을 찾는데 필요한 디스크 페이지 입출력 횟수

3.1 질의 처리 비용모델

질의 처리 비용모델을 수립하기 전에 비용모델에 사용되는 파라미터들을 〈표 1〉과 같이 정의한다. n_D는 질의 처리 당시의 문서 데이터베이스에 저장된 문서의 총 개수를 의미한다. $df(W)$는 키워드 W의 문서빈도수로서 전체 문서들 중에서 키워드 W가 포함된 문서의 수를 의미한다. $ps(W)$는 역색인 파일에서 키워드 W가 포함된 모든 문서의 식별자와 가중치를 구하기 위하여 키워드 W의 포스팅 리스트를 처음부터 끝까지 순차적으로 검색하는 비용으로서 포스팅 리스트 전체를 읽기 위해 필요한 디스크 페이지 입출력 횟수로 측정한다. $pa(W)$는 주어진 문서에 대한 키워드 W의 가중치를 구하기 위하여 키워드 W의 포스팅 리스트에서 주어진 문서 식별자가 포함된 포스팅을 찾는 비용으로서 포스팅 리스트를 탐색하는데 필요한 디스크 페이지 입출력 횟수로 측정한다.

키워드	포스팅 개수	포스팅 리스트 디스크 페이지 개수	포스팅 리스트 포인터
수출	1000	8	●
무역	1500	12	●
경제	2000	17	●
…	…	…	…

〈그림 2〉 역색인 파일의 키워드 디렉토리 예

비용 파라미터들의 값을 구하기 위하여 역색인 파일의 키워드 디렉토리를 〈그림 2〉와 같이 유지한다고 가정한다. 즉, 각 키워드마다 그 키워드의 포스팅 리스트를 가리키는 포인터와 더불어 포스팅의 개수 그리고 포스팅 리스트가 저장된 디스크 페이지의 개수를 유지한다. 그러면 포스팅의 개수가 바로 키워드의 문서 빈도수를 의미하므로 $df(W)$의 값을 정확하게 구할 수 있고, 포스팅 리스트가 저장된 디스크 페이지의 개수로부터 $ps(W)$의 값을 정확하게 구할 수 있다. 그러나 $pa(W)$는 포스팅 리스트를 탐색하는 방법에 따라 그 값이 달라진다. 본 논문에서는 포스팅 리스트의 구체적인 탐색 방법과 그에 따른 $pa(W)$의 값을 추정하는 방법에 대해서는 논의하지 아니한다.

정의 1 키워드 선택률

키워드 W의 선택률, $sel(W)$는 임의의 문서가 키워드 W를 포함할 확률로 정의하고 다음과 같이 구한다.

$$sel(W) = \frac{df(W)}{n_D} \quad \square$$

n개의 키워드 $W_1, W_2, \cdots, W_n$ 으로 이루어진 확장 불리언 질의를 $W_1, W_2, \cdots, W_n$ 순서로 처리할 때의 질의 처리비용, $C(W_1, W_2, \cdots, W_n)$은 다음과 같다.

$$C(W_1, W_2, \cdots, W_n) =$$

$$ps(W_1) + \sum_{i=2}^{n} (n_D \times \prod_{j=1}^{i-1} sel(W_j) \times pa(W_i)) \quad (1)$$

식(1)의 비용모델을 살펴보면, 첫 번째 키워드에 대해서는 포스팅 리스트를 한번 검색하는 비용이 들고, 두 번째 키워드에 대해서는 첫 번째 키워드가 포함된 문서의 개수만큼 두 번째 키워드의 포스팅 리스트를 탐색하는 비용이 든다. 세 번째 키워드에 대해서는 첫 번째 키워드와 두 번째 키워드가 모두 포함된 문서의 개수만큼 세 번째 키워드의 포스팅 리스트를 탐색하는 비용이 든다. 따라서 i 번째 키워드에 대해서는 첫 번째부터 i-1 번째까지의 키워드들이 모두 포함된 문서의 개수만큼 i 번째 키워드의 포스팅 리스트를 탐색하는 비용이 든다. 따라서 질의 처리비용이 키워드 처리 순서에 따라 달라진다.

3.2 질의 최적화 알고리즘

식(1)의 비용 모델은 Ibaraki(1984)가 발전한 ASI 성질을 만족한다. 즉, 식(1)의 비용 모델을 다음과 같이 재귀적으로 표현할 수 있다.

$$C(\Lambda) = 0$$
for null query Λ

$$C(W) = pa(W)$$
for query with a single keyword

$$C(S_1\ S_2) = C(S_1) + T(S_1) * C(S_2)$$
for query with keyword sequences S_1, S_2

여기서

$T(\Lambda) = 1$

$$T(S) = (n_D \times \prod_{W_i \in S}^{i-1} sel(W_j)$$

그리고 S에 대한 순위, $rank(S)$를 다음과 같이 정의하면

$$rank(S) = \frac{T(S) - 1}{C(S)}$$

다음의 보조정리 1이 성립한다.

보조정리 1

키워드 배열(sequence) A, B, U, V가 있을 때 $rank(U) \leq rank(V)$이면 $C(AUVB) \leq C(AVUB)$가 성립하며 그 역도 성립한다.

(증명)

$C(AUVB) = C(A) + T(A)C(U) + T(A)T(U)C(V) + T(A)T(U)T(V)C(B)$이므로

$C(AUVB) - C(AVUB)$

$= T(A)(C(V)(T(U)-1) - C(U)(T(V)-1))$

$= T(A)C(U)C(V)(rank(U) - rank(V))$

□

어떤 비용 함수가 보조정리 1을 만족하면 ASI 성질을 가진다고 말한다(Monma 1979). 따라서 식(1)의 비용 모델은 보조정리 1을 만족하므로 ASI 성질을 가진다. 그리고 ASI 성질을 가지는 비용 모델에 대해서는 순위에 기반한 질의 최적화가 가능하다(Ibaraki 1984). 본 논문에서는 이에 기반하여 각 키워드에 대한 순위로서 순차검색 역비용지수와 탐색 비용지수를 정의한다.

정의 2 키워드 순차검색 역비용지수

키워드 W의 순차검색 역비용지수 $r_s(W)$를 다음과 같이 정의한다.

$$r_s(W) = \frac{pa(W)}{sel(W)}$$

□

키워드 W의 순차검색 역비용지수 $r_s(W)$는 키워드 W의 포스팅 리스트 탐색 비용 $pa(W)$가 클수록 그리고 키워드 W의 선택률 $sel(W)$가 작을수록 커진다.

정의 3 키워드 탐색 비용지수

키워드 W의 탐색 비용지수 $r_a(W)$를 다음과 같이 정의한다.

$$r_a(W) = \frac{pa(W)}{1 - sel(W)}$$

□

키워드 W의 탐색 비용지수 $r_a(W)$는 키워드 W의 포스팅 리스트 탐색 비용 $pa(W)$와 키워드 W의 선택률 $sel(W)$가 클수록 커진다.

위에서 정의한 순차검색 역비용지수와 탐색 비용지수는 모두 Ibaraki(1984)가 정의한 순위와 같은 구조를 가지고 있으므로 이들을 이용하면 보조정리 1에 기반 하여 최적 질의 처리 방법을 도출할 수 있다.

정리 1 최적 질의 처리 방법

n개의 키워드, $W_1, W_2, \cdots, W_n$으로 이루어진 확장 볼리언 질의를 처리하는 최적 질의 처리 방법은 다음과 같은 순으로 키워드를 처리하는 것이다.

(1) 첫 번째 키워드는 순차검색 역비용지수 r_s의 값이 가장 큰 것을 선택한다.

(2) 두 번째 키워드부터는 탐색 비용지수 r_a의 값이 작은 순서대로 선택한다.

(증명)

(1) 첫 번째 키워드의 경우

최적 질의 처리 방법이 $W_1, W_2, \cdots, W_n$순으로 키워드를 처리하는 것임에도 불구하고 $r_s(W_1) < r_s(W_2)$가 되는 W_1과 W_2가 존재한다고 가정하면, W_1과 W_2의 위치를 서로 바꾸었을 때의 질의 처리비용이 원래 순서대로 처리했을 때의 비용보다 더 작게 되어 원래 순서가 최적이라는 주장이 모순이 됨을 보인다.

이를 위해 먼저 원래 순서와 바꾼 순서의 질의 처리비용 차이를 구해보면 다음과 같다.

$$\begin{aligned} & C(W_1, W_2, \cdots, W_n) - C(W_2, W_1, \cdots, W_n) \\ &= ps(W_1) + n_D \times sel(W_1) \times pa(W_2) \\ &\quad - ps(W_2) - n_D \times sel(W_2) \times pa(W_1) \\ &= n_D \times sel(W_1) \times pa(W_2) - n_D \times sel(W_2) \\ &\quad \times pa(W_1) + ps(W_1) - ps(W_2) \qquad (2) \end{aligned}$$

그런데 식(2)에서 n_D의 값이 충분히 크면 $ps(W_1) - ps(W_2)$는 무시할 수 있으므로

$$\begin{aligned} & C(W_1, W_2, \cdots, W_n) - C(W_2, W_1, \cdots, W_n) \\ &\simeq n_D \times sel(W_1) \times pa(W_2) \\ &\quad - n_D \times sel(W_2) \times pa(W_1) \end{aligned}$$

양변을 $n_D \times sel(W_1) \times sel(W_2)$로 나누면

$$\begin{aligned} & \frac{C(W_1, W_2, \cdots, W_n) - C(W_2, W_1, \cdots, W_n)}{n_D \times sel(W_1) \times sel(W_2)} \\ &= \frac{pa(W_2)}{sel(W_2)} - \frac{pa(W_1)}{sel(W_1)} = r_s(W_2) - r_s(W_1) \end{aligned}$$

그런데 가정에 의해 $r_s(W_2) - r_s(W_1) > 0$이고 $n_D \times sel(W_1) \times sel(W_2) > 0$이므로 $C(W_1, W_2, \cdots, W_n) - C(W_2, W_1, \cdots, W_n) > 0$이 된다. 즉, $C(W_1, W_2, \cdots, W_n) > C(W_2, W_1, \cdots, W_n)$이 되므로 원래 순서가 최적이라는 주장과 모순이 된다. 따라서 최적 질의 처리 방법은 항상 $r_s(W_1) \geq r_s(W_2)$ 이다.

(2) 두 번째 키워드부터의 경우

최적 질의 처리 방법이 $W_1, \cdots, W_i, W_{i+1}, \cdots, W_n$ 순으로 키워드를 처리하는 것임에도 불구하고 임의의 i에 대하여 $r_a(W_i) > r_a(W_{i+1})$가 되는 W_i와 W_{i+1}이 존재한다고 가정하면, W_i와 W_{i+1}의 위치를 서로 바꾸었을 때의 질의 처리비용이 원래 순서대로 처리했을 때의 비용보다 더 작게 되어 원래 순서가 최적이라는 주장이 모순이 됨을 보인다.

이를 위해 먼저 원래 순서와 바꾼 순서의 질의 처리비용 차이를 구해보면 다음과 같다.

$$\begin{aligned} & C(W_1, \cdots, W_i, W_{i+1}, \cdots, W_n) \\ &\quad - C(W_1, \cdots, W_{i+1}, W_i, \cdots, W_n) \\ &= n_D \times \prod_{j=1}^{i-1} sel(W_j) \times pa(W_i) \\ &\quad + n_D \times \prod_{j=1}^{i-1} sel(W_j) \times sel(W_i) \times pa(W_{i+1}) \\ &\quad - n_D \times \prod_{j=1}^{i} sel(W_j) \times pa(W_{i+1}) \end{aligned}$$

$$- n_D \times \prod_{j=1}^{i-1} sel(W_j) \times sel(W_{i+1}) \times pa(W_i)$$

$$= n_D \times \prod_{j=1}^{i-1} sel(W_j) \times pa(W_i) \times (1-sel(W_{i+1}))$$

$$- n_D \times \prod_{j=1}^{i-1} sel(W_j) \times pa(W_{i+1}) \times (1-sel(W_i))$$

양변을 $n_D \times \prod_{j=1}^{i-1} sel(W_j) \times (1-sel(W_i)) \times (1-sel(W_{i+1}))$로 나누면

$$\frac{C(W_1,\cdots,W_i,W_{i+1},\cdots,W_n) - C(W_1,\cdots,W_{i+1},W_i,\cdots,W_n)}{n_D \times \prod_{j=1}^{i-1} sel(W_j) \times (1-sel(W_i)) \times (1-sel(W_{i+1}))}$$

$$= \frac{pa(W_i)}{1-sel(W_i)} - \frac{pa(W_{i+1})}{1-sel(W_{i+1})}$$

$$= r_a(W_i) - r_a(W_{i+1})$$

그런데 가정에 의해 $r_a(W_i) - r_a(W_{i+1}) > 0$ 이고 $n_D \times \prod_{j=1}^{i-1} sel(W_j) \times (1 - sel(W_i)) \times (1 - sel(W_{i+1})) > 0$ 이므로 $C(W_1, \cdots, W_i, W_{i+1}, \cdots, W_n) - C(W_1, \cdots, W_{i+1}, W_i, \cdots, W_n) > 0$ 이 된다. 즉, $C(W_1, \cdots, W_i, W_{i+1}, \cdots, W_n) > C(W_1, \cdots, W_{i+1}, W_i, \cdots, W_n)$이 되므로 원래 순서가 최적이라는 주장과 모순이 된다. 따라서 최적 질의 처리 방법은 항상 $r_a(W_i) \leq r_a(W_{i+1})$이다.

□

위의 증명에서는 바로 이웃한 두 키워드에 대해서만 증명하였으나, 키워드들이 r_a값의 크기 순으로 나열되어 있으면 이웃하지 않은 두 키워드에 대해서도 성립한다. 즉, 이웃하지 않은 두 키워드, W_i와 $W_j (j > i+1)$에 대하여 $r_a(W_i) > r_a(W_j)$이 성립하면 이웃한 두 키워드, W_i와 W_{i+1}에 대해서도 $r_a(W_i) > r_a(W_{i+1})$이 성립한다. 따라서 이웃한 경우만 증명해도 일반성을 잃지 않는다. r_s의 경우도 마찬가지이다.

4 질의 최적화 실험

본 장에서는 절의 1에서 제시한 질의 처리 방법이 최적 질의 처리 방법임을 실험을 통하여 검증 해 본다. 또한 질의 최적화를 수행하는 경우와 수행하지 않는 경우의 질의 처리비용을 서로 비교해 봄으로써 질의 최적화의 중요성을 입증한다. 질의 처리비용은 수식 (1)의 비용모델에 의하며 질의 최적화를 수행하지 않는 경우에는 질의에 명시된 키워드 순서대로 질의를 처리한다고 가정한다.

〈표 2〉는 실험에 사용된 세 개의 키워드 W_1, W_2, W_3와 각 키워드에 대한 선택률, 포스팅 리스트 순차검색 비용, 포스팅 리스트 탐색 비용, 순차검색 역비용지수, 탐색 비용지수 등을 보여주고 있다. 키워드 W_1은 문서 빈도수가 작은 것을 선택하였고 키워드 W_2와 W_3은 문서 빈도수가 높은 것을 선택하였다. 키워드 선택률이 높으면 문서 빈도수가 높으므로 포스팅 리스트가 길어져 포스팅 리스트 순차검색 비용이 커진다.

포스팅 리스트 탐색 비용은 별도의 탐색 구조를 사용하지 않으면 포스팅 리스트 순차검색 비용과 같고, 사용하면 달라진다. 키워드 W_1과 W_2는 별도의 탐색 구조를 사용하지 않고 키워드 W_3은 별도의 탐색 구조를 사용한다고 가정한다. 따라서 키워드 W_1과 W_2의 포스팅 리스트 탐색 비용은 포스팅 리스트 순차검색 비용과 같고, 키워드 W_3의 포스팅

〈표 2〉 실험에 사용된 키워드

키워드	선택률	포스팅 리스트 순차검색 비용(ps)	포스팅 리스트 탐색 비용(pa)	r_s	r_a
W_1	0.1	10	10	100	11
W_2	0.8	100	100	125	500
W_3	0.9	120	7	8	70

n_D = 1,000,000

리스트 탐색 비용은 포스팅 리스트 순차검색 비용보다 작다. 〈표 2〉에서 보는 바와 같이 키워드 W_3의 포스팅 리스트 순차검색 비용은 키워드 W_2의 포스팅 리스트 순차검색 비용보다 더 크지만 별도의 탐색 구조를 사용함으로써 키워드 W_3의 포스팅 리스트 탐색 비용은 키워드 W_2의 포스팅 리스트 탐색 비용보다 더 작다. 한편, 본 실험에서는 저장된 총 문서의 개수 n_D를 1,000,000으로 가정하였다.

〈표 3〉은 세 개의 키워드 W_1, W_2, W_3를 가진 질의를 처리할 수 있는 모든 방법과 그 방법에 대한 질의 처리비용을 보여주고 있다. 이 중 질의 처리비용이 가장 작은 질의 처리 방법은 W_2, W_1, W_3순으로 처리하는 것이고, 가장 큰 방법은 W_3, W_2, W_1순으로 처리하는 것이다. 이 둘의 질의 처리비용을 비교해 보면 질의 처리비용의 차이가 매우 크므로 최소 비용의 질의 처리 방법을 찾는 질의 최적화가 중요하다는 것을 알 수 있다.

정리 1에 의하여 구한 질의 처리 방법이 실제로 최적 질의 처리 방법임을 〈표 2〉와 〈표 3〉을 통해 검증해 볼 수 있다. 정리 1에 의하면 첫 번째 키워드는 순차검색 역비용지수 r_s의 값이 가장 큰 키워드가 선택되므로 〈표 2〉에서 r_s값이 가장 큰 W_2가 선택된다. 그리고 두 번째 키워드부터는 나머지 키워드들 중에서 탐색 비용지수 r_a의 값이 작은 키워드 순으로 선택되므로 〈표 2〉에서 W_1의 r_a값이 W_3의 r_a값보다 더 작으므로 W_1, W_3순으로 선택된다. 따라서 W_2, W_1, W_3순으로 처리하는 것이 최적 질의 처리 방법이며 〈표 3〉을 통해 이것을 확인할 수 있다.

〈표 3〉 질의 처리 방법과 비용

질의 처리방법	질의 처리비용
W_1, W_2, W_3	10,560,010
W_1, W_3, W_2	9,700,010
W_2, W_1, W_3	8,560,100
W_2, W_3, W_1	12,800,100
W_3, W_1, W_2	18,000,120
W_3, W_2, W_1	97,200,120

5 결 론

본 논문에서는 역색인 파일을 이용한 확장 불리언 질의 최적화를 위해 비용에 기반한 체계적인 질의 최적화를 시도하였다. 확장 불리언 질의를 처리하는 방법은 질의를 구성하는 키워드의 처리 순서에 따라 여러 가지가 있을 수 있으므로 이 중에서 최소 비용을 가지는 질의 처리 방법을 구하는 질의 최적화가 필요하다. 본 논문에서는 이를 위해 질의 최적화 알고리즘을 제시하고 이의 최적성을 증명하였다. 그리고 실험을 통하여 질의 최적화를 수행할 경우와 수행하지 않을 경우의 질의 처리 비용 비교를 통해 질의 최적화의 중요성을 입증하였다.

비용 기반 질의 최적화를 위해서는 질의 처리비용 모델이 필요하다. 본 논문에서는 역색인 파일을 사용하여 확장 불리언 질의를 처리할 경우의 비용 모델을 수립하였다. 그리고 비용모델을 기반으로 하여 키워드에 대한 순차검색 역비용지수와 탐색 비용지수를 정의하고 이들을 이용하여 질의 최적화 알고리즘을 도출하였다. 역색인 파일을 이용한 불리언 질의 최적화 문제는 결국 최적 키워드 처리 순서를 구하는 문제임을 비용 모델을 통해 알 수 있고 이 문제는 메이터베이스 분야의 질의 최적화 문제 중 최적 조인 순서를 구하는 문제와 구조적으로 유사하다. 따라서 본 논문에서는 이 분야의 연구 결과를 이용하여 문제를 해결하였다.

확장 불리언 질의는 대부분의 정보검색 시스템에서 널리 사용되고 있으나 검색 성능 향상을 위한 질의 최적화가 제대로 이루어지지 않았다. 특히, 비용에 기반한 질의 최적화는 본 논문에서 처음으로 시도하였다고 사료된다. 실험 결과, 본 논문에서 제시한 질의 최적화 알고리즘은 실제로 최소 비용의 질의 처리 방법을 구하였고 질의 최적화를 하지 않을 경우와 비교하였을 때 그 성능이 월등히 우수함을 보였다. 따라서 오늘날 그 중요성이 증대되고 있는 정보검색시스템의 질의 처리 성능 향상에 본 논문이 큰 기여를 할 것으로 기대한다.

참 고 문 헌

Chaudhuri, S. and K. Shim. 1999. "Optimization of Queries with User-Defined Predicates." *ACM Trans. on Database Systems*, 24(2) : 177-228.

Cutting, D. and J. Pedersen. 1990. "Optimizations for Dynamic Inverted Index Maintenance." *Proc. Intl.* Conf. on Information Retrieval. ACM SIGIR. 405-411.

Elmasri, R. and S. B. Navathe. 2000. *Fundamentals of Database Systems*. Redwood City : the Benjamin/Cummings Publishing Company.

Faloutsos, C. 1985. "Access Methods for Text." *ACM Computing Surveys*, 17(1) : 49-74.

Frakes, W. B. and R. Baeza-Yates. 1992. *Information Retrieval - Data*

Structures & Algorithms. Englewood Cliffs, NJ: Prentice Hall.

Hanani, Michael Z. 1977. "An Optimal Evaluation of Boolean Expressions in an Online Query System." *Communications of the ACM*. 20(5): 344-347.

Ibaraki, T. and T. Kameda. 1984. "On the Optimal Nesting Order for Computing N-Relational Joins." *ACM Trans. on Database Systems*. 9(3): 482-502.

Jones, K. S. and P. Willett. 1997. *Readings in Information Retrieval*, Morgan Kaufmann Publishers.

Kaszkiel, M. and J. Zobel. 1998. "Term-ordered Query Evaluation versus Document-ordered Query Evaluation for Large Document Databases." *Proc. ACM SIGIR '98*. ACM SIGIR, 343-344.

Korfhage, R. R. 1997. *Information Storage and Retrieval*. Wiley Computer Publishing.

Lee, J. H., W. Y. Kim, M. H. Kim, and Y. J. Lee. 1993. "On the Evaluation of Boolean Operators in the Extended Boolean Retrieval Framework." *Proc. ACM SIGIR '93*. ACM SIGIR, 291-297.

Liu, Jane W. S. 1976. "Algorithms for Parsing Search Queries in Systems with Inverted File Organization." *ACM Trans. on Database Systems*. 1(4): 299-316.

Moffat, A. and J. Zobel. 1996. "Self-Indexing Inverted Files for Fast Text Retrieval." *ACM Trans. on Information Systems*. 14(4): 349-379.

Monma, C. and J. B. Sidney. 1979. "Sequencing with Series-Parallel Precedence Constraints." *Mathematics of Operations Research*. 4(3): 215-224.

Salton, Gerard. 1988. *Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer*, Addison -Wesley.

Selinger P. G., M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. 1979. "Access Path Selection in a Relational Database Management System." *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, 23-34.

Tomasic, A. and H. Garcia-Molina. 1993. "Query Processing and Inverted Indices in Shared-Nothing Text Document Information Retrieval Systems." *VLDB Journal*. 2: 243-275.

Witten, I. H., A. Moffat, and T. C. Bell. 1994. *Managing Gigabytes - Compressing and Indexing Documents and Images*, New York: Van Nostrand Reinhold.

Zobel, J., A. Moffat, and K. Ramamohanarao. 1998. "Inverted Files Versus Signature Files for Text Indexing." *ACM Trans. on Database Systems*. 23(4): 453-490.