대용량 텍스트 데이터베이스를 위한 효율적인 2단계 합성 요약 화일 방법

유 재 수[†] · 강 형 일[††]

요 약

본 논문은 대용량의 텍스트 문서를 효율적으로 처리하기 위해 단어 분별도(term discrimination) 개념을 이용한 2단계 합성 요약화일 방법(THM)을 제안한다. 또한 보다 더 나은 검색성능을 위해 2단계 합성 요약 화일 방법에 고분별력 단어들의 유사성에 의해 유사한 요약들을 함께 결집하는 Yoo가 제안한 요약결집 방법을 적용한다. 검색 시간,부가 저장 공간의 측면에서 제안된 2단계 합성 요약화일 방법(THM)의 성능 분석 모델을 제공하고 기존의 방법들 즉, 비트 슬라이스 방법(BM), 2단계 요약화일 방법(TM), 합성 방법(HM)들과 성능 평가를 수행한다. 성능 비교결과 일치하는 레코드의 수가 160이하일 때 100,000개의 대용량 데이터베이스에서 제안된 THM이 검색 성능면에 있어서 가장 좋은 성능을 보인다.

An Efficient Two-Level Hybrid Signature File Method for Large Text Databases

Jae Soo Yoo[†] · Hyungil Kang[††]

ABSTRACT

In this paper, we propose a two-level hybrid signature file method(THM) to efficiently deal with large text databases that use a term discrimination concept. In addition, we apply Yoo's clustering scheme to the two-level hybrid signature file method. The clustering scheme groups similar signatures together according to the similarity of the highly discriminatory terms so that we may achieve better performance on retrieval. The space-time analytical model of the proposed two-level hybrid method is provided. Based on the analytical model and experiments, we compare it with the existing methods, i.e. the bit-sliced method(BM), the two-level method(TM), and the hybrid method(HM). As a result, we show that THM achieves the best retrieval performance in a large database with 100,000 records when the number of matching records is less than 160.

1. 서 론

정보 관리 및 검색은 오랫동안 컴퓨터 시스템의 주요 연구 분야중의 하나이다. 전통적인 데이터베이스 시스템은 정형화된 데이타를 그 처리 대상으로 하였다. 그러나 과학적 데이터베이스[1], 사무 자동화 시스템[2], 멀티미디어 데이터베이스 등과 같은 최근의 응용들은 기본적으로 텍스트, 화상, 음성, 애니메이션 등과 같은 비정형적이고 복잡한 데이타의 처리를 요구한다. 이들 가운데 일반적으로 텍스트는 대부분의 응용분야에서 거의 공통적으로 이용되기 때문에 다

※본 연구는 학술진흥재단 1995년도 연구비 지원(과제번호:04-E-0049)에 의하여 수행되었음.
† 정 회 원:충북대학교 전기전자공학부 교수
†† 준 회 원:목포대학교 전산통계학과 석사과정
논문접수:1996년 7월 18일, 심사완료:1996년 10월 28일

중매체 데이타를 다루는데 매우 중요한 역할을 한다. 한편 텍스트에 대한 빠른 검색을 지원하기 위해 요약 화일 기법이 제안되었으며 이 기법은 텍스트에 대한 효율적인 저장 공간 사용뿐만아니라 효율적인 검색을 지원하기 때문에 현재 널리 연구되고 또한 이용되고 있다. 요약 화일 접근 기법은 각 문서에 대한 요약 정보를 요약 화일에 저장하고 질의 처리시에는 데이타 화일을 검사하기 전에 요약화일을 검사하여 질의를 만족할 가능성이 없는 문서들을 제거하여 문서검색 시간을 감소시킨다.

이들 가운데 Roberts[3]가 제안한 비트슬라이스 방법은 요약 화일 접근양을 줄이기위해 모든 요약을 해당 문서 순서대로 저장하는 대신 요약의 비트 순서대로 저장하는 방법을 이용한다. 두번째로 Sacks-Davis [4]가 제안하는 2단계 요약화일 방법은 대용량의 문서를 갖는 데이터베이스에서 빠른 검색을 지원하기 위해 문서 단위의 문서 요약 뿐만 아니라 문서의 블럭 단위의 집합을 위한 블럭 요약을 유지한다. 마지막으로 Falotsos와 Jagadish[8]는 비트 슬라이스 요약 화일 방법을 개선하기 위해 Zip's 법칙[7]에 기반을 둔 합성 방법을 제안하여 적은 비용의 부가 저장 공간으로 좋은 검색 성능을 보였다.

본 논문에서는 대용량의 텍스트 문서를 포함하는 최근의 응용들을 효율적으로 처리하기 위해 2단계 합성접근 방법을 제안한다. 제안된 2단계 합성 접근 기법은 Zip's 법칙에 근거하여 질의에는 자주 사용되나 데이타 화일의 구성 비율이 낮은 고분별력 단어들을 구별하여 빠른 검색을 지원하는 역화일로 구성하고 데이타 화일의 구성 비율이 높지만 질의에는 자주 사용되지 않는 저분별력 단어들을 적은 저장 공간을 사용하는 요약 화일로 구성한다. 또한 검색 성능을 향상시키기 위해 고분별력 단어들의 유사성에 의해 유사한 요약을 함께 결집하는 Yoo가 제안한 요약결집 방법인 CWD기법[11]을 사용한다. 제안된 방법의 성능을 평가하기 위해, 검색 성능과 부가 저장 공간 측면에서 분석적 비용 모델을 제시한다. 분석적 비용 모델과 실험을 통하여 100,000개의 도서 문서를 구성하는 데이타 화일에 근거하여 제안된 방법과 기존의 요약 화일 방법들의 성능을 비교한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 요약화일의 개념을 개괄적으로 설명하고 성능평가에 사용되는 각각의 방법들에 대해 알아보며 3장에서는 새로 제안한 2단계 합성 접근 방법을 제안하며 제안된 방법에 적용한 요약 결집 알고리즘을 설명한다. 4장에서 제안된 방법과 기존의 요약 화일 방법과의 성능 평가를 수행하기 위한 분석모델을 제시하고 5장에서 성능 비교 결과를 제시한다. 6장에서는 요약 화일 방법들을 검색 시간과 부가 저장 공간을 동시에 고려한 성능 평가 결과를 제시하고 7장에서 본 논문의 결론을 맺는다.

2. 요약화일

텍스트 검색 기법의 연구에서 Falotsos[2]는 텍스트 데이타와 정형화된 데이타에 적절한 효율적인 접근 기법으로 요약 화일 방법을 제안하였다. 사실상 요약 화일은 데이타 화일에 있는 레코드(또는 텍스트)를 요약화하여 저장한 화일로서 요약은 레코드를 검색할 색인 단어로 구성된 하나의 비트열(bit string)이다. 질의를 처리할때는 데이타 화일의 레코드를 검사하기 전에 먼저 레코드 요약을 검사하여 질의를 만족할 가능성이 없는 문서들을 제거함으로써 검색시간을 줄인다. 요약 화일 방법은 레코드 요약을 만들기위해 중첩 부호화(superimposed coding) 방법을 사용한다 [9]. 한 레코드가 n개의 색인단어로 구성되었을 때 각각의 단어는 해쉬 방법을 통하여 비트 스트링으로 변화되고 한 레코드 요약은 n 개의 단어 요약들을 비트별 중첩(bitwise ORing)하여 구성한다.

예를 들어 어떤 문서가 2개의 단어 "Text", "Retrieval"을 포함하고 있을 때 다음은 중첩 부호화 방식을 사용하여 문서 요약을 만드는 과정을 나타낸다.

문서 D=(Text, Retrieval)

키워드	단어 요약
Text	0010 0100
Retrieval	1000 1000
문서 요약	1010 1100

요약 화일 방법은 문서속의 단어의 갯수가 요약의 길이에 영향을 미치지 않으므로 문서속에 있는 단어

의 갯수가 가변적인 경우도 아주 자연스럽게 지원한다.

요약 화일 방법에서 검색은 먼저 문서 요약을 구성하는 방법과 동일하게 질의 요약을 만든 후 요약 화일에서 질의 요약을 비트별로 포함하는 레코드 요약들을 선택한다. 마지막으로 실제 질의를 만족하는 레코드 요약을 추출한다. 그러나 요약 화일을 통과하여 검색이 이루어지는 문서들 중에 실제 질의를 만족하지 않는 경우가 있는데 이를 오류매치(false match)라고 한다. 오류매치가 발생할 가능성은 질의 요약과 레코드 요약에 존재하는 비트들의 수에 관계한다.

그 동안 요약 화일의 효율적인 접근 구조를 제공하기 위하여 많은 연구가 진행되어왔다. 그중 일반적인 3가지 기법이 1) 비트 슬라이스 기법, 2) 다단계 인덱스 기법, 3) 단어 분별도에 기반을 둔 합성 기법이다. 먼저 Roberts[3]는 요약 화일 전체를 접근하는 것을 피하기 위하여 요약의 순서가 아닌 요약의 비트 순서로 요약을 저장하는 비트 슬라이스 기법을 제안하였다. 질의가 질의 요약의 크기 b에서 w개의 "1"을 포함한다면 각 요약의 관련된 w비트만 조사하면 된다. 이 방법은 보통 $w \ll b$이므로 검색시간을 많이 향상시킬 수 있다. 검색 시간을 줄이기 위한 두번째 방법은 다단계 요약 화일을 기반으로 한다. 매우 큰 데이타 화일에서는 비트 슬라이스 요약 화일 방법은 여전히 비용이 많이 든다. Sacks-Davis[4]는 Roberts가 제안한 비트 슬라이스 표현 방법과 Pfalt의 다단계 요약 화일 방법[12]을 결합하여 2단계 요약 화일(TM)을 제안하였다. 2단계 요약 화일 방법은 문서 요약 화일에 블럭 요약 화일을 더한 것이다. 2단계 요약 화일 구조에서 질의를 처리하는 과정은 먼저 블럭 요약 화일을 검색하여 레코드 요약 화일의 접근할 블럭을 결정한 후 레코드 요약 화일의 해당 블럭을 검색하여 만족하는 레코드를 최종적으로 결정한다.

Faloutsos와 Jagadish[8]은 비트 슬라이스 방법을 확장하기 위해 합성 방법(HM)을 제안하였다. 단어 분별도 개념을 사용하여 저분별력 단어를 위해서는 요약 화일 구조를 사용한 반면 고분별력 단어를 위한 별도의 포스팅 화일을 구성하였다. 80-20규칙이 지켜질 경우 합성 방법은 비트 슬라이스 방법과 비교해 볼 때 적은 비용의 부가 저장 공간으로 검색 효율을 높인다. 〈표 1〉은 언급된 3가지 요약 화일 기법에 따른 관련 연구를 나타낸다.

〈표 1〉 요약 화일의 3가지 기법에 따른 분류

		비트 스트링 기법	비트 슬라이스 기법
일단계 기법	1 경로	Files & Huskey[5]	Roberts[3]
	합성 기법	N/A	Faloutsos & Jagadish[8]
다단계 기법	1 경로	Sacks-Davis & Ramamohanarao [6]	Sacks-Davis & Ramamohanarao [6]
	합성 기법	N/A	N/A

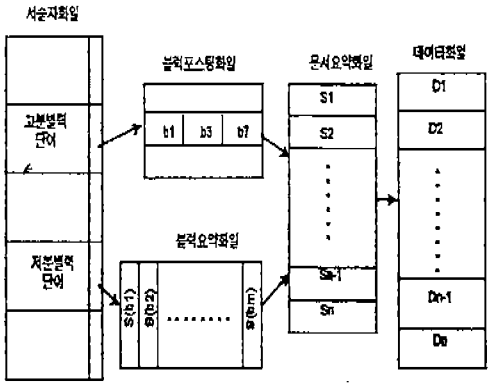
3. 2단계 합성 접근 방법

Faloutsos와 Jagadish에 의해 제안된 합성 방법은 적은 비용으로 좋은 검색 성능을 제공하지만 대용량의 문서를 처리하는 데에 여전히 검색 성능이 떨어지는 단점을 가지고 있다. 이러한 문제점을 해결하기 (그림 1)과 같은 2경로 요약화일 방법과 역화일 방법으로 구성된 새로운 기법을 제안하였다. Zip's법칙에 근거하여 질의에는 자주 사용되나 데이타 화일의 구성 비율이 낮은 고분별력 단어들의 특별한 처리를 한다. 즉 효율적인 검색을 하기위해 고분별력 단어를 빠른 검색을 지원하는 역화일로 구축한다. 또한 검색 성능을 향상시키기 위해 고분별력 단어들의 유사성에 의해 비슷한 문서 요약들을 함께 결집하는 CWD 결집 기법[11]을 사용한다.

처음 단계인 인덱스 화일을 구축하기 위하여 우수한 검색성능을 제공하는 B^+ 트리구조를 사용하며 비록 역화일 방법이 많은 부가저장공간을 차지하지만 그러한 문제는 블럭 포스팅 화일에 고분별력 단어만을 저장하기 때문에 큰 문제가 안된다. Zip's법칙에 근거하여 데이타 화일에 존재하는 약 20%의 단어들이 사용자 질의에 약 80%를 차지한다는 80-20 규칙이 지켜질 경우 합성요약화일 기법은 20%의 단어에 대해서만 블럭 포스팅 화일을 구축하므로 기존 역 화일의 약 1/5의 부가 저장 공간만을 필요로 한다. 역화일 기법을 사용하는 또 다른 동기는 인덱스 화일에 실제적인 단어를 저장하기 때문에 오류 매치를 완전히 피할 수 있기 때문이다. 또한 요약 화일 방법이 효

율적인 저장 공간 사용을 제공하기 때문에 저분별력 단어인 경우 블럭 요약 화일을 사용한다.

질의를 처리할 때 처음에 인덱스 화일을 탐색하고 주어진 단어가 어느 분별력 단어에 속하는지 알아 다음의 블럭 디스크립터의 주소를 찾아낸다. 다음 블럭이 질의와 일치하는지 확인하기 위해 블럭 포스팅 화일과 블럭 요약 화일을 접근한 후 두 검색의 결과를 모은다. 마지막으로 문서를 확인하기 위해 문서 요약 화일 안에 있는 일치하는 블럭으로부터 문서 요약을 조사한다.



(그림 1) 2단계 합성 요약 파일 기법

제안된 2단계 합성 요약 화일 방법의 검색 성능을 향상시키기 위해 결집 방법을 적용한다. 결집방법의 사용은 요약화일을 효율적으로 접근하기 위한 또 하나의 방법이다. 사실상 결집 방법은 도서 검색 시스템 구축에 있어서 매우 큰 관심을 받는 방법이다[10]. 결집 기법에서는 문서의 결집을 위해 유사한 문서가 함께 분류되고 물리적으로 함께 저장이 된다. 그러나 결집 알고리즘을 사용하여 구성된 화일 구조에서는 삽입 연산을 쉽게 처리할 수 없으며 찾고자 하는 문서가 있더라도 검색하지 못하는 경우가 발생한다. 위의 2가지 문제점을 해결하기 위해 유사한 레코드 보다는 비슷한 레코드 요약들을 결집하는 결집 방법들을 Yoo[11]가 제안하였다. Yoo의 CWD 결집 방법은 요약 결집을 쉽고 효율적으로 제공하기 위하여 단어 분별 개념을 이용한다. 이를 위해 CODASYL 데이타베이스의 관련 튜플들을 결집하기 위해서 경험적 그래프 충돌 기법을 사용한 Malmquist's알고리즘을 적용하여 다음과 같은 요약 결집 알고리즘을 고안하였다[11].

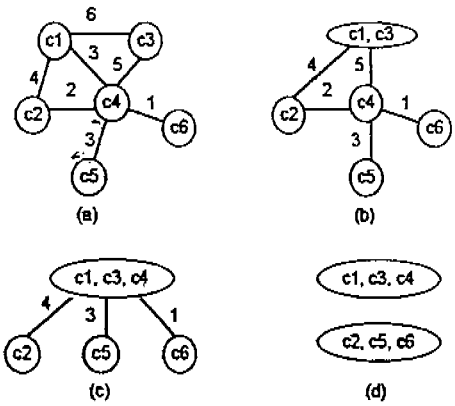
알고리즘: 레코드 요약 결집
입력:
R: 레코드 집합 $\{r_i | r_i \in R, 1 \le i \le n\}$
E: 두 집합 사이의 관련 정도의 집합,
$\{e_{ij} | e_{ij} \in E, 1 \le i, j \le n\}$
PT: 레코드의 고분별력 단어 집합의 모임들
$\{pt_{ij} | pt_{ij} \in PT_i, 1 \le i, j \le s\}$
Limit: 블럭당 미리정해진 레코드 요약의 갯수

출력:
C: 유사한 레코드 요약들을 저장하는 결집들의 집합

변수:
$w(e_{ij})$: e_{ij}의 관련 정도,
즉 공통 고분별력 단어 개수
$num(C_i)$: 결집 C_i내에 포함되는 레코드의 개수

처리:
/* 변수의 초기화 */
$C=\{c_i | c_i = r_i, 1 \le i \le n\}$,
all $w(e_{ij})=0$, all $num(c_i)=1$;
/* 결집 생성 */
while ($E \neq \phi$) {
최고의 관련 정도를 지닌 E의 하나의 원소
e_{ij}찾기:
$E=E-\{e_{ij}\}$;
if ($num(c_i) + num(c_i) \le Limit$) {
i가 j보다 작은 c_i와 c_j를 c_i에 합병:
$num(c_i) = num(c_i) + num(c_j)$;
For each $e_{kj} \in E$ such that $1 \le k \le n$
and $k \neq i \neq j$,
if ($e_{kj} \in E$) then
$E=E-\{e_{kj}\}+\{e_{ki}\}$;
else $E=E-\{e_{kj}\}$,
$w(e_{kj})$ max($w(e_{kj})$, $w(e_{ki})$);
}
}
end:

예를 들어, 알고리즘이 6개의 레코드에 대한 그림 2(a)와 같은 일련의 초기 결집을 구성한다고 가정하자. 그림 2(a)의 각 노드는 한 개의 레코드를 구성하는 결집을 나타내며 가중치를 갖는 에지는 두 결집간의 유사성의 정도 즉 공통 고분별력 단어의 개수를 나타낸다. 그림 2(b)와 (c)는 계속적으로 에지의 가중치가 가장 큰 두 개의 결집을 합병하여 더 큰 결집을 만드는 과정을 보여주고 있다. 알고리즘의 마지막 단계는 고정된 크기를 초과하지 않는 결집들을 묶어 다른 하나의 결집을 형성하게 된다.



(그림 2) 요약 결집과정

4. 성능 분석

제안한 2단계 합성 접근 기법의 검색 시간, 부가 저장 공간의 측면에서의 성능 분석을 위해서 r개의 임의로 선택된 토큰을 포함하는 예상되는 블럭 개수를 e(r, m, n)으로 정의하며, 이때 m은 블럭당 토큰의 개수를, n은 블럭의 개수를 나타낸다. 마찬가지로 m*n개의 토큰이 그들의 유사성에 의해서 n개의 블럭내에 결집 되었을 때 r개의 토큰을 포함하는 예상되는 블럭 개수는 c(r, m, n)으로 정의한다. 제안하는 2단계 합성 접근 기법을 분석하기 위한 입력 및 설계 인자들은 〈표 2〉와 같으며 이 인자들을 통해 성능 평가를 수행한다.

• R_{BM}, R_{TM}, R_{HM}, R_{THM} : 사용자 질의어가 평균적으로 $n*\alpha$의 고분별력 단어와 $n*(1-\alpha)$의 저분별력 단어로 구성된 질의를 검색할 때, 각각의 기법들의 디스크 접근 횟수

• O_{BM}, O_{TM}, O_{HM}, O_{THM} : 각각의 기법들의 부가 저장 공간 (페이지 수)

(1) 검색 시간

• R_{if} : HM과 THM의 인덱스 화일 접근을 위한 디스크 접근 횟수, $[\log_d V]$ 이고
$d = (P-t)/(w+t)$

• R_{pf} : HM의 포스팅 화일의 접근을 위한 디스크 접근 횟수, $[e(E, N_r, N_s)t/P]$

• R_{ppf} : THM의 블럭 포스팅 화일의 접근을 위한 디스크 접근 횟수, $[c(E, N_r, N_s)t/P]$

• R_{bsf} : THM의 블럭 요약 화일의 접근을 위한 디스크 접근 횟수, $k_S * [N_S/P]$

• R_{rsf} : TM의 레코드 요약 화일의 접근을 위 한 디스크 접근 횟수,
$(e(A, N_r, N_s)+N_s*F_s)(1+N_r*F_r)$

• R_{rsfc} : 결집된 THM의 레코드 요약 화일의 접 근을 위한 디스크 접근 횟수
$(c(A, N_r, N_s)+N_s*F_s)(1+N_r*F_r)$

• R_{sf} : BM의 요약 화일의 접근을 위한 디스크 접근 횟수, $k_r*[N/P]$

• R_{df} : 데이타 화일을 위한 디스크 접근 횟 수, A

제시된 부가적인 인자를 통한 검색 성능은 다음과 같다.

$$R_{BM} = n*R_{sf}+R_{df}$$
$$R_{TM} = n*R_{bsf}+R_{rsf}+R_{df}$$
$$R_{HM} = n*R_{if}+n*\alpha*R_{pf}+n*(1-\alpha)*R_{sf}+R_{df}$$
$$R_{THM} = n*R_{if}+n*\alpha*R_{ppf}+n*(1-\alpha)*R_{bsf}$$
$$+\alpha*R_{bsf}+(1-\alpha)*R_{rfscc}+R_{df}$$

(2) 부가 저장 공간

• O_{if} : HM과 THM의 인덱스 화일을 위한 부가 저장 공간, $[(w+t)*V/P]$.

• O_{pf} : HM의 포스팅 화일을 위한 부가 저장 공 간, $[U*e(E, N_r, N_s)*t/P]$

- O_{ppf}: THM의 블럭 포스팅 화일을 위한 부가 저장 공간, $[U * c(E, N_r, N_s) * t/P]$
- O_{bsf}: TM의 블럭 요약 화일을 위한 부가 저 장 공간, $[b_s * N_s/P]$
- O_{rsf}: TM의 레코드 요약 화일을 위한 부가 저장 공간, N_S
- O_{sf}: BM의 요약 화일을 위한 부가 저장 공 간, $[b_r * N/P]$

〈표 2〉 입력과 설계 인자

기호	정 의
N	레코드의 총 갯수
P	페이지 크기(비트 갯수)
A	질의를 만족하는 레코드의 갯수
E	질의를 만족하는 레코드의 평균 갯수
N_s	레코드 요약화일의 크기
N_r	블럭당 레코드 요약의 갯수($N = N_s * N_r$)
b_s, b_r	요약의 크기(블럭과 레코드 요약화일)
k_s, k_r	한 단어에서 1로 세트되는 비트 갯수
F_s, F_r	요약의 false match 확률
n	질의에 있는 단어의 수
α	고분별력 단어가 사용자 질의에 사용될 확률
V	데이타 화일내의 비중복 단어의 총수
U	데이타 화일내의 비중복 고분별력 단어의 총수
s	레코드당 모든 단어들의 평균 갯수
r	레코드내의 고분별력 단어의 평균 갯수
t	포인터 크기(바이트 수)
w	한 단어의 평균 갯수(바이트 수)
h	역화일로 사용된 B-tree의 높이
c	역화일내의 한 단어의 평균 체인 길이

제시된 부가적인 인자를 통한 부가 저장 공간의 크기는 다음과 같다.

$$O_{BM} = O_{sf}$$
$$O_{TM} = O_{bsf} + O_{rsf}$$
$$O_{HM} = O_{if} + O_{pf} + O_{sf}$$
$$O_{THM} = O_{if} + O_{ppf} + O_{bsf} + O_{rsf}$$

제안하는 2단계 합성 접근 기법의 효율성을 보이기 위해 다른 요약 화일 기법과 검색 성능, 부가 저장 공간의 관점에서 제안된 분석적 비용 모델을 사용하여 비교한다. 이와같은 성능 비교를 위해 각각의 입력과 설계 인자들의 값은 〈표 3〉에 제공되며 Sacks-Davis의 데이터베이스에 근거하여 구축된다[4]. 데이타 화일이 100,000개의 문서로 구성되고 단어 분별도의 비율이 80-20 규칙을 따를 때 분석 비용 모델을 통한 각각의 방법의 성능 평가를 수행한다.

성능 평가결과는 검색 성능에 있어서 주된 차이점이 c로 표시된 요약 결집 이득으로부터 야기된다. 여기서 c, 즉 $c(E, N_r, N_s)$는 N개의 문서요약이 $N = N_r * N_s$인 고분별력 단어의 유사성에 따라 N_s 블럭내에 결집되었을 때, 질의 요약을 만족하는 매칭 레코드 요약 A를 담고 있는 예상 블럭 개수이다. 데이터베이스 예에서 나타내듯이 A가 N_s보다 상대적으로 작을 때

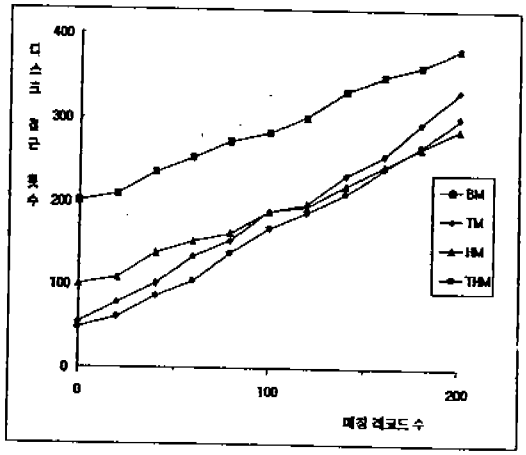
〈표 3〉 입력 및 설계 인자값들

M / P	BM	TM	HM	THM
b_s	N/A	17341	N/A	14173
k_s	N/A	10	N/A	10
b_r	1471	1471	1471	1229
k_r	10	10	10	10
β	N/A	2.3	N/A	3.43
V	N/A	N/A	313437	313437
U	N/A	N/A	189381	189381
N_s	N/A	3760	N/A	3760
N_r	N/A	22	N/A	22
r	N/A	N/A	17	17
F_r	1/N	$1/(19 * N_r)$	1/N	$1/(29 * N_r)$
F_s	1/N	$1/N_s$	1/N	$1/N_s$
N	100,000			
L	1295			
P	32768			
s	185			
n	1-5			
A	0-14000			
E	383			
α	80-20			
w	7			
t	8			
h	5			
c	0.8			

$e(A, N_r, N_s)$는 대체로 A와 같다. 그것은 문서요약에 접근하기 위한 매칭 문서와 같은 블럭 접근의 숫자, 다시 말해 블럭 접근의 최대치가 필요하다는 것을 의미한다. 그러므로 여기에서 $e(A, N_r, N_s) \geq c(A, N_r, N_s)$임을 증명할 수가 있다. $c(A, N_r, N_s) = A/\beta$라 하면 β는 요약 결집의 정도를 가리키는 결집 요소이다. β가 증가함에 따라 더 많은 레코드 요약들이 동일 블럭내에 결집된다. 따라서 검색 성능의 향상을 가져온다.

한편 (그림 3)은 요약 결집도 β에 근거하여 2단계 합성 요약 화일 방법과 기존의 요약화일 방법들의 검색 성능을 디스크 접근 횟수로 나타낸 것이다.

(그림 3)에서 보여주는 바와 같이 본 논문에서 제안하는 2단계 합성 요약 화일(THM)은 기존의 다른 방법들에 비해 평균 약 37%의 검색 성능을 향상시켰다. 또한 검색 성능에서 HM과 THM 같은 합성 방법이 BM과 TM보다 성능이 우수함을 확신할 수 있었다. 부가 저장 공간 측면에서는 HM과 THM이 BM과 TM보다 약 13% 더 많은 부가 공간을 요구한다.

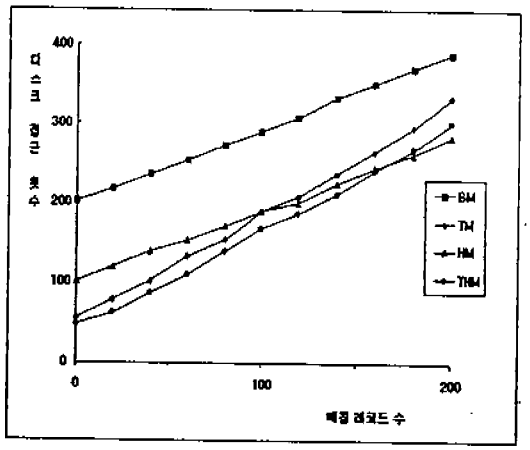


(그림 3) 분석적 비용 모델을 통한 검색 성능 평가

5. 실험적 고찰 결과

4가지 요약화일 방법에 근거한 접근 기법들의 성능 평가를 위하여 BM, TM, HM, THM의 방법을 구현하고 도서 문서에 근거하여 실험을 수행하였다. 실험에 사용된 도서 문서 레코드는 Author, Title, School, Degree, Date, pp, Adviser, Source, Publication, Subject, Abstract와 같은 11개의 애트리뷰트로 구성되고 평균 214개의 키워드로 이루어진다. 실험을 위해 〈표 3〉에서 제시된 입력과 설계 인자값들 그대로 사용하였다.

100,000개의 도서문서로 구성된 데이터베이스로 실험을 수행하였는데 여기에서 br는 각방법에 있어서 레코드 요약의 길이를 나타내고 bs는 2경로 방법의 블럭 요약의 길이를 나타낸다. 이러한 매개 변수들에 의해서 성능 평가를 수행한다. 검색 성능은 매칭 문서의 단어들에 블럭 접근의 수에 의해 측정된다.



(그림 4) 실험을 통한 검색 성능 평가

실험 결과를 보여준다. 일반적으로 대화식 환경에서는 매칭 문서의 갯수가 200이하이다[11]. 결과적으로 (그림 4)에서 보여주는 바와같이 합성 방법인 HM과 THM이 BM과 TM보다 우수함을 알 수 있다. 게다가 매칭 문서의 갯수가 160인 점에서 THM과 TM이 교차하는 것을 볼 수 있다. THM이 매칭 문서의 갯수가 160이하일 때 기존 요약 화일 방법들에 비해 훨씬 더 우수한 검색 성능을 보인다. 결론적으로 THM이 100,000개의 도서 문서로 구성된 대용량 데이터베이스에서 다른 방법들에 비해 20-55%의 검색 성능을 향상시켰다.

〈표 4〉는 부가 저장 공간 측면에서는 4가지 접근방법의 실험적인 결과를 비교하여 보여 준다. 결과적으로 HM과 THM이 BM과 TM보다 약 10-18% 더 많은 부가 공간을 요구한다.

〈표 4〉 부가 저장 공간의 실험적 결과

저장공간 방법	대용량 데이타베이스	
	블럭의 수	수가 저장 공간
BM	4685	9%
TM	7101	14%
HM	10028	19%
THM	13198	25%
데이타 화일	53566	100%

분석 비용 모델의 정확성을 증명하기 위한 분석 비용 모델를 통한 결과와 실험적 결과에 근거한 에러율을 계산하기 위해 다음과 같은 수식을 사용한다. 여기에서 E와 T는 각각 이론적 결과와 실험적 결과를 나타낸다.

$$\text{Error Rate} = [\text{Max}(T, E) - \text{Min}(T, E)] / \text{Max}(T, E)$$

100,000개의 문서로 실험한 결과 검색 측면에서 요약화일 방법들의 에러율이 약 13% 정도로 나타났다. 그러나 (그림 3)과 (그림 4)에서 나타난 바와같이 각 요약 화일 방법들의 검색 성능의 특성은 분석적 비용 모델과 실험을 통한 결과가 거의 일치한다. BM과 TM과 같은 1경로 방법들의 부가 저장 공간 측면에서 에러율은 극히 적었다. 그러나 THM과 같은 2경로 방법들은 분석 비용 모델과 실험을 통한 에러율이 1경로 방법들에 비해 좀 더 많았다. 이와 같은 이유는 분별력 단어들의 비율과 역화일의 높이, 포스팅 화일의 평균 체인 길이를 정확하게 평가하기 어렵기 때문이다. 결론적으로 실험을 통해 분석 결과와 실험적 결과가 대체로 일치한다는 것을 확인할 수 있었다.

6. 고 찰

일반적으로, 검색시간과 부가 저장 공간 사이에는 상호 대조적이다. 즉 하나의 접근 방법이 검색 성능이 우수하면 다른 방법에 비해 부가 저장 공간을 더 많이 요구한다. 그러므로 접근 방법의 검색 성능과 부가 저장 공간의 일반적인 특징을 제공할 필요가 있다. O_{METHOD}는 METHOD라고 불리우는 접근 기법이 요구하는 부가 저장공간을 나타낸다고 할 때 성능평가 결과 4가지 접근 방법들의 부가 저장 공간 요구량이 $O_{THM} > O_{HM} > O_{TM} > O_{BM}$임을 알 수 있다. 즉 BM이 가장 적은 부가 저장 공간을 요구하는 반면 THM의 부가 저장 공간 요구량이 가장 많이 요구함을 알 수 있다.

반면에 검색 성능이 데이터베이스의 크기와 매칭 문서의 수에 의존하기 때문에 검색 성능 면에 있어서 일반적인 특징을 알기에는 쉽지않다. 여기에서 R_{METHOD}가 METHOD라고 불리는 접근 방법의 검색 측면에서 디스크 접근의 횟수를 나타낸다고 하자. 한 예로 매칭 문서의 갯수가 160이하인 대용량 데이터베이스에서는 검색 효율성의 측면에서 접근 방법들의 순서가 $R_{THM} < R_{HM} < R_{TM} < R_{BM}$임을 알 수 있다.

검색 성능과 부가 저장 공간을 동시에 고려한 가장 효율적인 기법을 선택하기 위해 다음과 같은 비용 공식을 제시한다.

$$CE_{METHOD} = w_R * R_{METHOD}/N_R + w_O * O_{METHOD}/N_O,\ N_R = R_{BM}\text{이고 } N_O = O_{DF}$$

여기에서 N_R과 N_O는 같은 축에서 검색 시간과 부가 저장 공간을 제공하기 위한 기준이다. N_R은 비트 슬라이스 방법(BM)이 가장 떨어지기 때문에 R_{BM}을 사용한다. N_O는, 데이타화일이 부가저장 공간의 크기를 결정하는 요소로 크게 좌우하기 때문에 O_{DF}, 데이타 화일의 저장 공간을 사용한다. 또한 w_R과 w_O는 각기 검색 시간과 부가 저장 공간 크기의 상대적인 중요성을 나타내는 상대적 가중치이다. 예를 들어, 대용량 데이터베이스에서 매칭문서의 갯수가 50일 때 각각의 접근 기법들의 비용 효율을 계산해 보았다. 정보 검색 환경에서 성능 평가시 검색 시간이 부가 저장 공간 보다 중요시되기 때문에 $w_R = 0.7$, $w_O = 0.3$이라고 가정하였다.

1) $CE_{BM} = 0.7 * 240/240 + 0.3 * 4685/53566 = 0.73$

2) $CE_{TM} = 0.7 * 111/240 + 0.3 * 7101/53566 = 0.36$

3) $CE_{HM} = 0.7 * 138/240 + 0.3 * 10028/53566 = 0.46$

4) $CETHM = 0.7 * 96/240 + 0.3 * 13198/53566$

$=0.35$

여기에서 2단계 합성 요약 화일 방법이 비용면에서 가장 효율적인 반면 비트슬라이스 방법이 가장 많은 비용을 요구한다. 또한 본 논문에서 제안한 2단계 합성 방법(THM)이 2단계 방법(TM)과 거의 같은 비용 효율치를 나타냄을 알 수 있다. 그러나 THM은 TM이 갖고있는 문제점 즉, 범위 질의 처리, 동의어 처리를 하지 못하는 문제점들을 해결한다. 그러므로 제안된 방법은 경로 선택 문제, 범위 질의 처리, 동의어 처리를 지원하고 탐색 성능이 좋고 저장 장소도 많이 요구하지 않는 적당한 방법이다.

7. 결 론

본 논문은 기존의 정적 요약 화일 방법들의 검색 성능 향상을 위하여 Zip's법칙에 근거한 2단계 합성 접근 기법(THM)을 제안하였다. 즉, 효율적인 검색 성능을 위해 고분별력 단어를 빠른 검색을 지원하는 역화일로 구축하였으며 검색면에서 더 나은 성능을 수행하기 위해 고분별력 단어들의 유사성에 의해 유사한 요약을 함께 결집 하는 CWD 결집기법을 사용한다. THM의 검색시간과 부가 저장 공간 측면에서 제안된 THM의 분석 모델을 제공하고 기존의 방법들 즉, BM, TM, HM들과 성능을 비교 하였다. 결과적으로 THM이 대용량 데이터베이스에서 20~55%의 검색 성능을 향상시켰다. 즉, 검색 성능에서 HM과 THM과 같은 합성 방법이 BM과 TM 보다 성능이 우수하였으며, 매칭 문서의 갯수가 적을 때 대용량 데이터베이스(100,000 레코드)에서는 THM이 HM보다 성능이 우수하였다. 결과적으로 합성 방법들 중에서 데이터베이스의 크기와 매칭 문서의 갯수에 의해 가장 적당한 방법을 선택 사용하면 된다. 마지막으로 향후 연구로는 WISS, MEDLARS, STAIRS 등과 같은 실질적인 정보 검색 시스템에 본 논문에서 제안한 THM을 적용해 보는 것이 필요하다.

참 고 문 헌

[1] Zezula P. et al., "Dynamic Partitioning of Signature Files", ACM Trans. on Information Systems, vol. 9, no. 4, pp. 336-369, Oct. 1991.

[2] Faloutsos C., "Access Methods for Text", ACM Computing Survey, vol. 17, no. 1, pp. 49-74, Mar. 1985.

[3] Roberts C. S., "Partial-Match Retrieval via the Method of Superimposed Codes", Proceedings IEEE, vol. 67, no. 12, pp. 1624-1642, Dec. 1979.

[4] Sacks-Davis R. et al., "Multikey Access Methods Based on Superimposed Coding Techniques", ACM Trans on Database System, vol. 12, no. 4, pp. 655-696, Dec. 1987.

[5] Files J. R. and Huskey H. D., "An Information Retrieval System Based on Superimposed Coding", Proceeding of AFIPS Fall Joint Computer Conference, pp. 423-432, 1969.

[6] Sacks-Davis R. and Ramamohanarao K., "A Two Level Superimposed Coding Scheme for Partial Match Retrieval", Information Systems, vol. 8, no. 4, pp. 273-280, 1985.

[7] Berra P. B. et al., "Computer Architecture for a Surrogate File to a Very Large Data/Knowledge Base", IEEE Computer, vol. 20, no. 3, pp. 25-32, Mar. 1987.

[8] Faloutsos C. and Jagadish H. V., "Hybrid Index Organizations for Text Databases", CS-TR-2621, Computer Science, Univ. of Maryland, 1991.

[9] Knuth D. E., "The Art of Computer Programming", vol. 3, pp. 506-549, 1974.

[10] Willett P., "Recest Trend in Hierarchie Document Clustering: A Critical Review", Information Processing and Management, vol. 24, pp 577-597, Nov 1988.

[11] J. S. Yoo and H. S. Choi, "Heuristic Clustering Methods for Efficient Signature File Construction", KISS Journal, vol. 22, no. 12, pp. 1625-1633, 1995.

[12] J. L. Pfaltz, W. J. Berman and E. M. Cagley, "Partial-Match Retrieval Using Indexed Descriptor Files", Communications of the ACM, Vol. 23, No. 9, Sept. 1980, pp. 150-168.

유 재 수

1980년 전북대학교 공과대학 컴퓨터공학과(학사)
1991년 한국과학기술원 전산학과(공학석사)
1995년 한국과학기술원 전산학과(공학박사)
1995년~1996년 8월 목포대학교 전산통계학과 전임강사
1996년 8월~현재 충북대학교 공과대학 전기전자공학부 전임강사
관심분야: 데이터베이스 시스템, 정보검색, 멀티미디어 데이터베이스, 분산 객체 컴퓨팅 등

강 형 일

1996년 목포대학교 전산통계학과(학사)
1996년 목포대학교 전산통계학과 석사과정
1996년 9월~현재 목표대학교 전산통계학과 조교
관심분야: 멀티미디어 데이터베이스, 정보검색, 저장구조 등