제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

# 멀티미디어 텍스트 데이터 검색을 위한 접근기법 연구

양창호, 정윤기, 이배호
전남대학교 컴퓨터공학과
e-mail : terabig@empal.com

# A Study on Access Control of the Multimedia Text Data Retrieval

Chang-Ho Yang, Yoon-Ki Jung, Bae-Ho Lee
Dept. of Computer Engineering, Chonnam National University

요 약

컴퓨터와 통신의 급속한 발전으로 인하여 하루에도 수십 기가바이트의 정보가 매일매일 업데이트 되고 있다. 하지만 이러한 유용한 정보의 증가에도 불구하고 우리가 사용의 어려움과 검색시간이 길어진다면 엄청난 정보의 낭비를 초래할 것이다. 멀티미디어 정보에 대한 접근은 데이터의 특성상 매우 신속해야 하므로 검색시간 또한 최소화되어야 한다. 하지만 대용량의 멀티미디어 데이터베이스에서 데이터 접근은 막대한 시간을 낭비할 소지가 다분하다. 멀티미디어 데이터 접근은 데이터베이스를 구성하는 여러 미디어에 대해 생성되는 메타데이터에 기본을 둔다. 또한 사용되는 인덱스 구조는 미디어, 메타데이터, 질의 형식에 기반을 두고 생성된다. 즉 인덱싱의 기법에 따라 탁월한 검색성능의 향상을 보일 수 있다. 본 논문에서는 멀티미디어 데이터중 텍스트 데이터 접근에 이용 가능한 여러 가지 인덱싱 기법들을 살펴보고 그에 따른 적용방법들을 제안한다.

## 1. 서론

최근 컴퓨터 하드웨어의 급속한 발전, 대용량 저장 장치의 출현, 컴퓨터 통신기술의 발달로 인해 다양한 형태의 대규모 데이터, 즉 멀티미디어 데이터를 처리하는 일이 가능하게 되었다.[3] 멀티미디어 데이터베이스 시스템이란 기존의 데이터베이스에 저장되어 있는 문자 정보나 숫자뿐만이 아니라 새로운 정보의 형태인 멀티미디어 데이터를 효율적으로 저장, 검색할 수 있는 기능을 갖춘 시스템을 말한다.[1][2][4]

통상적으로 텍스트 데이터에 대한 검색에는 불리언 질의와 키워드 검색을 들 수 있다. 불리언 질의는"(data or information) and retrieval and (not text)"의 형태로 불리언 연산자와 연결하여 검색할 수 있다. 몇몇 추가적인 연산자로는 "adjacent"나 "within n words"와 같은 명백한 의미를 지닌 연산자가 사용된다. 예를 들면 질의 "data within sentence retrieval"은 두 개의 단어 "data"와 "retrieval"을 가진 구문을 포함한 문서를 검색할것이다. 키워드 검색은 "data, retrieval, information"과 같은 키워드를 사용자가 입력하면 검색 시스템은 위의 키워드 중에서 가능한 많은 수의 키워드를 포함한 문서를 사용자에게 출력할 것이다. 또한 현재의 시스템은 전형적으로 전위(Prefix) 검색을 허용한다. 예를 들면 사용자가 "organ *"처럼 "organ"으로 시작하는 단어와 관련된 모든 단어를 찾으라는 "*" 명령어를 사용한다고 가정하자. 그러면 시스템은 사용자에게 organ으로 시작하는 문자 즉 "organs", "organization", "organism" 등을 찾을 것이다. 텍스트 데이터에 대한 검색을 위해 생성되는 텍스트 메

* 본 연구는 정보통신부 대학기초사업 연구비 지원사업에 의해 수행된 연구결과임

타데이터는 문서 내에서 발생하는 인덱스의 특성뿐만 아니라 문서에 대한 묘사로 구성된다.[3] 신속한 텍스트 데이터로의 접근을 제공하기 위해 메타데이터를 저장하기 위한 적절한 저장구조가 사용되어야 한다. 또한 텍스트 접근을 위한 인덱스 특성의 선택은 사용자 질의를 위한 적절한 문서를 선택하는데 도움을 준다.[4]

## 2. 인덱스 특성의 선택

문서빈도와 역 문서빈도는 인덱스 특성의 선택을 위해서 널리 이용되는 요소이다.[2] 문서빈도는 선택된 인덱스 특성이 나타나는 문서의 수이며 다음과 같이 표시된다.

$$df(\Phi_i) = |\{d_j \in D \mid ff(\Phi_i, d_j) > 0\}|$$

또한 역 문서빈도는 다음과 같이 정의된다.

$$idf(\Phi_i) = \log(\frac{n+1}{df(\Phi_i)+1})$$

통상적으로 인덱스 특성의 선택은 문서빈도 $df(\Phi_i)$가 특정한 상한 경계값 아래에서 나타나도록 한다. 이것은 인덱스 특성을 포함한 문서의 수를 적게 하여 검색과정에 소요되는 시간을 단축하기 위한 것이다. 또한 이것은 선택된 인덱스 특성 $(\Phi_i)$에 대한 역문서빈도 $idf(\Phi_i)$가 높음을 암시한다.

## 3. 풀 텍스트 스캐닝(Full Text Scanning)

풀 텍스트 특성 검색을 위한 단순한 알고리즘으로는 검색 특성에 해당하는 문자들을 문서 안의 모든 문자들과 비교하는 방법이 있다. 우선 문자 한 개에 대한 일치를 찾고 해당 문자가 일치하지 않을 경우에는 오른쪽으로 한 문자 옮겨서 다시 일치하는지를 알아본다. 이런 검색 방식은 문서 내에서 일치하는 특성이 발견되어지거나 문서의 끝에 도달할 때까지 계속되는 식으로 진행된다. 이 알고리즘은 무척 간단하지만, 하나의 인덱스 특성을 찾기 위한 비교 횟수가 너무 크다는 단점이 있다. 만약 검색 특성의 길이가 M이고 문서의 길이가 N이면 최악의 경우 O(M*N)의 비교가 필요하게 된다. 위의 알고리즘에 약간의 변경을 가하면 검색 속도가 크게 향상된다. 기본적인 아이디어는 찾고자 하는 특성에 대한 불일치가 발생했을 경우에 텍스트 포인터의 위치를 효율적으로 많이 이동시킬 수 있는 방법을 찾는다는 것이다. 이런 텍스트 검색을 위한 알고리즘들은 <그림 1>처럼 FSM(Finite State Machine)을 사용하여 설계될 수 있다. 특정 텍스트 스트링의 검색을 위한 FSM은 다음의 과정을 거쳐서 만들어 낸다.

· 진행함수의 정의 : 이 함수는 하나의 입력 심볼에 대한 FSM의 상태 변환을 정의한다. 특정 입력 심볼에 대하여 상태 변환이 정의되지 않았을 경우에 진행 함수는 실패로 보고된다.

· 실패 함수의 정의 : 이 함수는 진행 함수가 실패를 발생했을 때 불리어진다. 실패 함수는 실패 메시지가 발생했을 경우 상태의 변환에 대해서 정의한다. 이러한 실패 변환 이후 동일한 입력 심볼과 새로운 상태에 대한 진행 함수가 수행된다.

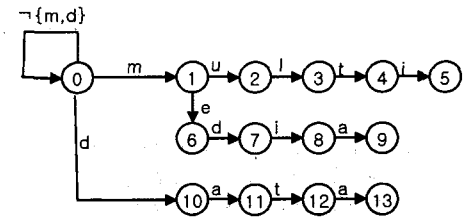


그림 1 키워드 검색을 위한 FSM의 예

· 출력 함수의 정의 : FSM은 출력 상태의 집합을 가지며, 출력 함수는 각각의 출력 상태에 대한 검색 키워드를 정의한다.

풀 텍스트 스캐닝(Full text scanning) 방법은 인덱스 파일과 같은 별도의 검색 정보가 저장될 필요가 없다는 장점을 가지고 있다.

## 4. 역 파일(Inverted file)

역 파일은 문서에 대한 검색 정보를 저장하기 위해 사용된다. 저장되는 검색 정보는 포스팅(posting) 집합과 인덱스 특성을 포함한다.

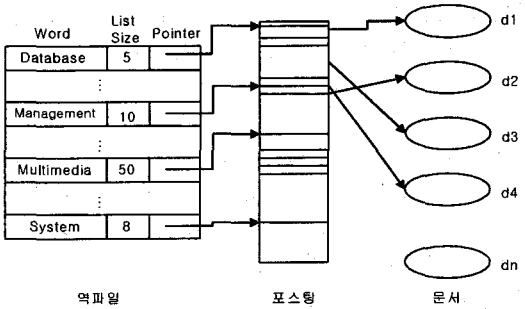


그림 2 역 파일 구조

포스팅은 인덱스 특성이 어떤 문서에서 나타나는 것에 대한 정보를 표시한다. 즉, 문서에 대한 포인터다. <그림 2>는 역 파일에 대한 구조를 단면적으로 보여주고 있다. 이러한 역 파일에 대한 접근은 단일 키(single key)에 기반을 두고 있으므로 인덱스 특성에 대한 효율적인 접근이 가능해야 한다. 인덱스 특성은 알파벳순으로 정열(sort)되거나, 해시 테이블(hash table) 형태로 저장되거나, 혹은 B-트리와 같은 복잡한 구조를 사용하여 저장되기도 하지만 여기서는 해시 테이블에 대해서만 다루기로 하겠다.
문자열로 표시되는 인덱스 특성을 해시 테이블 내의 위치로 바꾸기 위해서는 해시 함수가 적용되어야만 한다. <그림 3>은 특성 인덱스와 그에 해당하는 포스팅을 저장하기 위하여 사용되는 해시 테이블이다.

| | | | | |
|---|---|---|---|---|
| Multimedia | 011 | 010 | 001 | 100 |
| Database | 010 | 001 | 100 | 010 |
| Management | 001 | 100 | 010 | 001 |
| System | 011 | 011 | 101 | 110 |
| **Signature** | **011** | **111** | **111** | **111** |

그림 3 다수 애트리뷰트 검색을 위한 첨가 코딩

역 파일의 장점은 특성에 대한 빠른 접근 방법이 제공되므로 사용자 질의에 대한 응답시간의 감소가 가능하다는 것이다. 하지만 그에 반해 인덱스 특성과 문서의 수가 커질 때, 역 파일의 크기가 매우 커질 수 있다는 점이다.

## 5. 다수 애트리뷰트 검색

텍스트 문서를 찾기 위한 질의가 하나 이상의 특성으로 구성될 경우에는 역 파일이 아닌 다른 기법이 사용되어야 한다. 예를 들어 서명이 "Multimedia database management system"인 책을 찾는다고 가정하면, 이 질의에서는 'Multimedia', 'database', 'management', 'system'의 네 가지 키워드가 지정되어 있다. 해시 파일을 사용할 경우 각 애트리뷰트

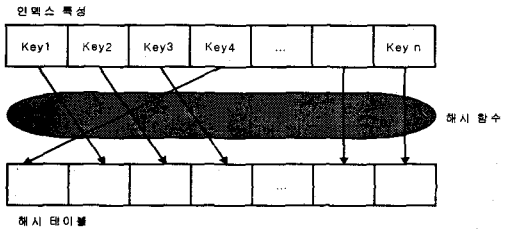


그림 4 역 파일을 위한 해시 테이블

는 일정한 길이를 가지는 비트 패턴(bit pattern)으로 변형되고 질의를 위한 signature를 만들기 위하여 4개의 비트 패턴은 OR(boolean OR) 연산자에 의해서 포개진다. <그림 4>는 이 네 개의 키워드로부터 질의 특성을 위한 signature를 얻는 과정을 보여준다.

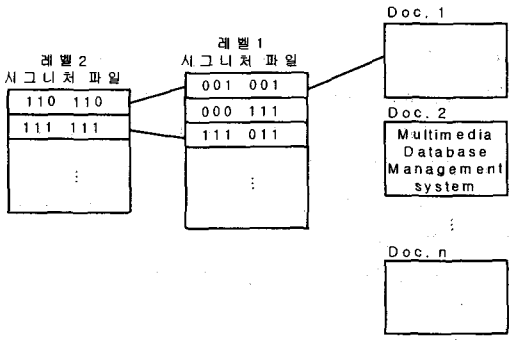


그림 5 Multi-Level Signature 파일

여기에서 signature의 크기는 12비트로 가정되었다. signature 값인 '011 111 111 111'은 인덱스 특성인 "Multimedia database management system"을 텍스트 문서에서 검색하기 위한 정보로서 사용되어진다. <그림 5>는 각각 6비트인 두 단계의 signature를 사용하는 방법이다.

## 6. 텍스트 문서의 클러스터링

유사한 문서들을 클러스터링하는 것은 검색 속도를 향상시킬 수 있다. 흔히 하나의 클러스터(Cluster)로 묶인 유사한 문서들은 동일한 질의에 연관되기 때문이다. 클러스터링 기법을 문서 대신 인덱스 특성에 적용할 수도 있다. 클러스터링 한다는 관점에서 살펴볼 때 문서나 인덱스 특성이나 검색 질의는 모두가 m-차원 공간에서의 점으로 표현된다. 이때 문서 기술자 $d_j=(a_{1,j}, \cdots, a_{m,j})$의 형태로 정의될 수 있다. 여기에서 m은 인덱스 특성의 수이며, a(i,j)는 각각의 특성과 관련된 가중치(weight)를 나타낸다. 만약 하나의 특성이 문서를 특징지으면 그 특성에 대한 가중치는 높고 반대의 경우 가중치는 낮아진다.
<그림 6>은 가중치를 이용한 문서의 클러스터링을 보여준다. 그림에서 클러스터 $\{c_1,\cdots, c_n\}$은 문서를 특징짓기 위해 사용되는 인덱스 특성의 집합니다. 예를 들면 c2는 인덱스 특성인 "Multimedia"가 나타나는 문서를 표현한다고 보면, 문서 d2, d3와 관련

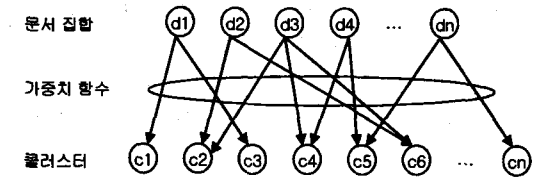


그림 6 가중치를 이용한 클러스터링

된 가중치는 두 개의 문서와 인덱스 특성인 "Multimedia" 와의 관계치를 묘사한다. 만약 문서 d3이 "Multimedia"와 별로 관계가 없다면 가중치 (d3, c2)는 매우 낮다.

다음의 가중치 함수들은 문서 클러스터링에서 자주 쓰이는 함수들이다.

- 이진 문서 기술자 : 인덱스 특성이 나타나면 1, 없으면 0
- 특성 빈도 : $ff(\phi_i, d_j)$
- 문서 빈도 : $df((\phi_i)$
- 역 문서 빈도 : $idf(\phi_i)$
- $ff(\phi_i, \phi_j)*R_j$ ($R_j$ : 문서 $j$를 위한 특성 관계 요소)

위에서 언급된 가중치 함수의 값은 문서 클러스터링을 위하여 계산된다. 그중에서 이진 문서 기술자, 문서 빈도, 역 문서 빈도 및 특성 빈도에 기반을 둔 가중치 함수는 인덱스 특성으로부터 직접 계산이 가능하다. 이진 문서 기술자는 인덱스 특성이 나타나는가 나타나지 않는가에 의해 평가된다.

## 7. 가중치 함수를 위한 학습 기반 접근방법

대부분의 학습 기반 방법은 확률론에 근거를 두고 있다. <그림 7>은 이러한 학습 기반 방법의 일반적인 원칙을 보여준다.

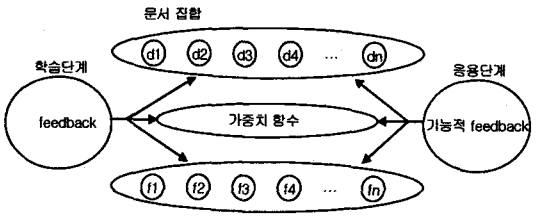


그림 7 클러스터링을 위한 학습 기반 방법

학습 기반 방법은 두 가지의 단계를 가진다. 첫 번째는 학습 단계로서 여러 개의 학습 질의를 이용하여 피드백 정보를 얻어낸다. 이러한 학습 질의들은 텍스트 접근을 위한 질의와 유사하며, 특정 문서 혹은 문서집합에 대해 적용될 수 있다. 문서를 선택하기 위한 두 가지 종류의 질의 사이의 관련성에 기반을 두고, 인덱스 특성이나 문서에 대하여 확률적인 가중치가 할당된다. 두 번째는 응용단계로서 학습 단계 동안에 얻어진 가중치를 기반으로 일반적인 질의에 대한 답을 구한다. 또한 관련된 가중치를 수정하기 위한 피드백 정보가 이러한 일반적인 질의로부터 유도될 수 있다.

## 8. 결론 및 사후과제

본 논문에서는 텍스트 데이터에 대한 접근 방법과 컨텐츠를 효율적으로 추출할 수 있는 처리 기법들을 제시하였다. 멀티미디어 정보들은 그 용량이 거대하여 접근과 사용상의 상당한 제약과 시간적 손실이 발생한다. 하지만 본 논문에서 이러한 제약들을 감소시킬 수 있는 여러 가지 방법들을 사용하여 용이한 컨텐츠의 접근이 가능하다. 멀티미디어 데이터는 텍스트뿐만 아니라 음성, 영상, 동영상 등 수없이 많은 이질적인 정보들이 존재한다. 그만큼 데이터가 커지면서 그에 따른 충분한 검색능력을 향상시키는 기법들을 꾸준히 연구해야 할 것이며, 이런 이질 정보들을 통합시키는 기법 또한 개발되어져야 할 것이다.

## 참고문헌

[1] A. Tomasic, H.G. Molina and K. Shoens, "Incremental Updates of Inverted Lists for Text Document Retrieval", Proceedings of ACM SIGMOD' 94, pp. 289-300, 1994.

[2] C. Faloutsos, "Indexing Multimedia Databases", Advanced Course on Multimedia Databases in Perspective, University of Twente, The Netherlands, pp. 239-278, 1995.

[3] Elmasri/Navathe, 데이터베이스 시스템, 생능출판사, pp. 139-183, 1998.

[4] 신동규/신영일, 멀티미디어 데이터베이스, 교보문고, 2000.

[5] H.V. Jagdish, "A Retrieval Technique for Similar Shapes", International Conference on Management of Data, SIGMOD'91, pp. 208-217, 1991.