한글 텍스트를 위한 요약 화일 기법의 설계

장 재 우
전북대학교, 컴퓨터공학과

Design of the Signature File Method for Hangul Text

Chang, Jae Woo
Department of Computer Engineering, Chonbuk National University

요 약

텍스트를 이용하는 새로운 데이타베이스 응용을 효율적으로 지원하기 위해 여러 가지 텍스트 검색 기법이 연구되었으며, 이러한 연구 가운데 효율적인 검색 기법으로 요약 화일 (signature file) 방법이 제안되었다. 그러나 이러한 연구는 모두 영문 텍스트를 위한 연구이며, 한글 텍스트를 위한 요약 화일 기법에 관한 연구는 거의 전무한 상태이다. 따라서 본 논문에서는 한글의 특성에 맞는 요약 화일 기법을 설계하고 아울러 제안한 기법의 실용성과 타당성을 검토한다.

I. 서 론

데이타베이스 시스템의 응용이 사무 자동화, 도서 관리, 멀티미디어 등의 분야로 확장됨에 따라, 비정형화된 데이타인 텍스트를 처리하는 확장된 데이타베이스 관리 시스템이 요구되었다 [4,9]. 한편 이러한 새로운 응용을 효율적으로 지원하기 위해 여러가지 텍스트 검색 기법들이 연구되었으며, 이러한 연구가운데 대표적인 방법으로 요약 화일 (signature file) 기법이 제안되었다 [3,5,6].

요약 화일 기법은 기존의 텍스트 검색 기법으로 널리 사용된 역 화일 (inverted file) 기법의 단점을 극복하기 위해 제안된 방법이다. 사실 역 화일 기법은 DIALOG, STAIRS, BRS, MEDLARS, ORBIT, LEXIS 등의 정보 검색 시스템 (information retrieval system) 에서 널리 사용된 텍스트 검색 기법이다 [14]. 그러나 역 화일 기법은 부가적으로 데이타 화일의 50-300% 에 해당하는 대용량의 저장 공간을 필요로 하는 단점이 있다. 이에 반해 요약 화일 기법은 검색 속도 면에서 역 화일 기법에 비해 다소 뒤떨어지나, 부가 저장 공간의 크기가 데이타 화일의 10% 내외로 상당히 적다는 점에서 대규모 데이타베이스를 필요로 하는 응용에 적합하다 [4,5].

요약 화일 기법은 텍스트를 저장한 데이타 화일 이외에 각 텍스트에 대한 요약을 별도의 요약 화일에 유지하는 방법이다. 질의 처리시 데이타 화일의 텍스트를 검사하기에 앞서 먼저 그 텍스트의 요약을 검사하여 질의를 만족할 가능성이 있는 텍스트만을 선택하는 방법이다. 이 기법의 효율성을 위해 Roberts 는 도치된 (bit-sliced) 요약 화일 방법을 제안하였고 [10], Sacks-Davis 는 블럭 요약 화일과 레코드 요약 화일로 구성된 2 단계 요약 화일 (two-level signature file) 방법을 제안하여 기존의 요약 화일에 비해 검색 성능이 향상됨을 보였다 [11,12,13]. 아울러, Chang 은 2 단계 요약 화일 방법에 역 화일 구조나 해쉬 테이블 같은 블럭 서술 화일을 결합한 2 경로 2 단계 요약 화일 (two-path two-level signature file) 방법을 제시하여 Sacks-Davis 의 연구를 확장하였다 [2].

최근 역 화일 기법 일변도로 나아가는 정보 검색 시스템의 텍스트 검색 기법이 요약 화일 기법으로 대체되고 있는 싯점이다. 이것의 가장 대표적인 예가 TITAN 시스템으로서 이 시스템은 요약 화일 기법중 가장 널리 알려진 2 단계 요약 화일 방법을 사용한 시스템으로, 검색 속도 면에서나 부가 저장 공간 면에서나 모두 우수한 것으로 알려져 있다 [1]. 이제 한국형 정보 검색 시스템을 구축하려는 여러가지 논의들이 오고가는 가운데 가장 시급히 해결해야 할 문제는, 한글 디쳐러스 (thesaurus) 구축, 한글 문서의 표준화, 효율적인 한글 텍스트 검색 기법 등이다. 이 가운데 가장 기초적이고 중요한 일은 한글 특성에 맞는 효율적인 텍스트 검색 기법에 관한 연구이다.

앞서 언급한 바와 같이, 최근 요약 화일 기법은 가장 효율적인 텍스트 검색 기법으로 알려져 있다. 그러나 불행이도 이러한 연구는 대개 영문 텍스트를 위한 연구이며 한글 텍스트를 위한 요약 화일 기법에 관한 연구는 거의 전무한 상태이다. 따라서 본 논문은 한글 특성에 맞는 효율적인 텍스트 검색 기법을 위해, 한글 텍스트의 특성을 분석, 파악하여 한글 텍스트를 위한 요약 화일 기법을 설계하고 아울러 설계된 방법의 실용성과 효율성을 실증하는 데 그 목적이 있다. 앞으로 정보 사회로의 급속한 이전에 따라 필수적으로 요구되는 한국형 정보 검색 시스템의 구축에 있어 그 하부 구조로서 필수 불가결한 요소는 빠른 검색 속도, 최소의 부가 저장 공간, 한글 특성에의 부합성 등을 갖춘 텍스트 검색 기법이라 할 때, 이러한 점은 한글 특성에 적합한 요약 화일 기법을 설계함으로써 달성될 수 있다.

II. 요약 화일 기법

일반적으로 사무 자동화, 도서 관리, 멀티미디어 등의 시스템에서는 대규모 데이타를 사용하며, 아울러 비정형화된 데이타 즉 텍스트에 대한 효율적인 처리가 필요하다 [3,4]. 이러한 응용을 위한 대표적인 텍스트 검색 방법인 역 화일 기법은, 검색 시간이 빠른 반면 데이타 화일의 50-300% 에 해당하는 부가 저장 공간이 필요하다는 단점이 있다 [4]. 한편 이에 대한 대안으로서 요약 화일 기법은 요약 화일의 크기가 데이타 화일의 불과 10% 를 차지한다는 점과 비정형화된 텍스트를 쉽게 처리한다는 점에서 이러한 환경에 적합하다.

요약 화일 기법은 각 레코드 (또는 텍스트) 에 대한 요약 (signature) 을 요약 화일에 저장하고, 데이타 화일을 검색하기에 앞서 요약 화일을 검색하여 질의를 만족할 가능성이 있는 레코드만을 선택 접근함으로써, 데이타 화일 검색 시간을 감소시키는 텍스트 검색 기법이다. 일반적으로 각 레코드의 요약은 단어 요약을 비트별로 중첩하여 (bitwise ORing) 구성한다 [8]. 각 단어에 대한 요약은 해싱 (hashing) 을 사용하여 구성하며 기존의 해싱 방법과는 달리 요약 화일에서 사용하는 해싱은, 해싱에 따른 결과 비트 스트링의 '1' 의 갯수가 일정하게 유지되는 방법이다. 다음은 레코드가 'database', 'text', 'retrieval' 의 3 개의 단어로 구성된 경우 레코드 요약을 만드는 과정을 나타낸다. 여기서 m 과 k 는 각각 레코드 요약의 크기와 한 단어에 할당된 '1' 의 갯수를 나타낸다.

```
해싱 H : m=12, k=2
H(database) : 0110 0000 0000
H(text)     : 0000 0000 0011
H(retrieval): 0000 1001 0010
----------------------------
레코드 요약 : 0110 1001 0011
```

한편 요약 화일 기법에서 질의의 선택 조건을 만족하는 레코드를 검색하는 방법은 다음과 같다. 먼저 질의로 부터 레코드 요약을 구성하는 방법과 같이 질의 요약을 만들고, 데이타 화일을 접근하기 전 단계로서 요약 화일을 접근하여, 질의 요약의 비트 패턴 (bit pattern) 을 포함하는 요약들을 추출한다. 이때 추출된 요약에 해당하는 레코드는 질의를 만족할 가능성이 있는 레코드로 간주하고, 이러한 레코드만 데이타 화일에서 최종적으로 검색하여 질의를 만족하는 레코드로 추출한다. 그림 1 은 4 개의 레코드로 구성된 데이타 화일과 4 개의 요약으로 구성된 요약 화일에서 질의에 대한 처리 과정을 보여준다.

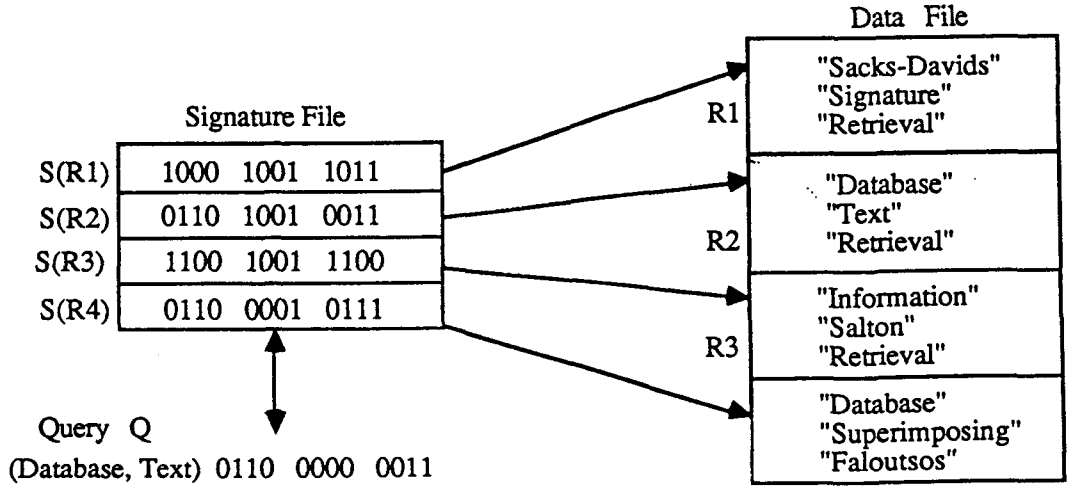

그림 1. 요약 화일의 질의 처리 과정

질의 단어 'database', 'text' 로 부터 질의 요약을 구성하고, 요약 화일의 각 레코드 요약 중에서 S(R2) 와 S(R4) 가 질의를 만족할 가능성이 있음으로 판정되고, 마지막으로 데이타 화일의 R2 와 R4 를 실제로 검색하여 R2 만이 질의를 만족함이 판명된다. 이때 질의를 만족할 가능성이 있지만 실제로 질의를 만족하지 않은 R4 는 false match 레코드라 한다.

요약 화일 기법에서 효율적인 검색을 위한 연구는, 데이타 화일을 검색하는 시간을 줄이는 연구와 요약 화일을 검색하는 시간을 줄이는 연구로 분류된다. 데이타 화일을 효율적으로 검색하기 위한 연구는 주로 false match 를 줄이기 위한 연구로서, 주어진 요약 화일 크기를 가지고 최소의 false match 를 위해 요약 생성에 필요한 최적의 '1' 로 되는 비트 개수를 구하는 것으로 달성된다 [3,6,10]. 이는 결국 불필요한 데이타 화일의 접근을 최소화시킨다. 두번째로 요약 화일을 효율적으로 접근하기 위한 연구는 요약 화일의 구조에 중점을 둔 연구로서, 비트 슬라이스 요약 화일 (bit-sliced signature file) 방법, 2 단계 요약 화일 (two-level signature file) 방법, 그리고 2 경로 2 단계 요약 화일 (two-path two-level signature file) 방법이 제안되었다.

(1) 비트 슬라이스 요약 화일 방법

비트 슬라이스 요약 화일 방법은 요약 화일 전체를 검색하는 것을 피하기 위하여, 요약 순서가 아닌 요약의 비트 순서로 요약을 저장하는 것으로, 모든 요약의 첫번째 비트들이 차례로 저장되고 다음 두번째 비트들이 순서적으로 저장되는 형태이다 [10]. 이때 요약의 비트 순서로 요약을 저장할 때의 열을 비트 슬라이스 (bit slice) 라 하고, 요약 순서로 요약을 저장할 때의 열을 비트 스트링 (bit string) 이라 한다. 비트 슬라이스 요약 화일 방법에서 레코드 요약을 검색할 때는 해당되는 비트 슬라이스만을 접근하므로 비트 스트링 방법보다 검색 속도 면에서 훨씬 우월함을 알 수 있다. 그러나 새로운 레코드를 삽입할 때는 모든 비트 슬라이스를 요약 화일에 추가해야 하므로 요약 화일의 재구성이 요구되는 단점이 있다.

(2) 2 단계 요약 화일 방법

Pfaltz 는 다수개의 요약들을 비트별 중첩한 요약을 다음 단계의 요약 화일에 저장하는 다단계 요약 화일 기법을 제안하였다. 기존의 요약 화일 기법이 요약 화일 전체를 검색하는 데 비하여, 다단계 요약 화일 기법은 최고 단계의 요약 화일만 완전 검색하기 때문에 보다 적은 시간내에 요약 화일을 검색할 수 있다. 한편 Sacks-Davis 는 비트 슬라이스 요약 화일 방법과 다단계 요약 화일 방법을 결합하여, 비트 슬라이스 표현 방식의 블럭 요약 화일과 비트 스트링 표현 방식의 레코드 요약 화일로 구성된 2 단계 요약 화일 방법을 제시하였다 [11,12,13]. 이때 블럭 요약 화일은 블럭내에 있는 레코드들을 구성하는 단어들의 요약을 블럭 단계에서 비트별 중첩하여 만든다. Sacks-Davis 는 2 단계 요약 화일이 대규모 데이타베이스에서 질의를 만족하는 레코드의 개수가 적을 때, 1 단계 요약 화일보다 검색 성능이 효율적임을 보였다 [13].

(3) 2 경로 2 단계 요약 화일 방법

Faloutsos 는 텍스트에는 질의에 자주 나타나는 고분별력 단어와 질의에 상대적으로 적게

나타나는 저분별력 단어가 존재하며, 고분별력 단어는 반대로 데이타 화일에는 적게 나타난다고 주장했다 [7]. 2 경로 2 단계 요약 화일 방법은 Faloutsos 의 고분별력 단어 개념을 이용하여, 기존 Sacks-Davis 의 2 단계 요약 화일 방법을 검색 시간 측면에서 개선하기 위하여 제안되었다 [2]. 즉 요약 화일보다 검색 성능이 우수한 역 화일이나 해쉬 테이블 등으로 고분별력 단어들을 위한 블럭 서술 화일을 구성하고, 레코드 요약 화일을 고분별력 단어의 관련성에 의해 결집 (clustering) 하였다. 만약 80-20 규칙이 지켜질 경우, 20% 의 고분별력 단어에 대하여 역 화일을 구성하고 나머지 80% 의 단어에 대하여 요약 화일을 구성한다. 이때 역 화일을 통하여 질의의 80% 를 처리하므로 2 단계 요약 화일 방법보다 검색 성능이 향상된 반면, 데이타 화일의 20% 만을 역 화일로 구성하기 때문에 기존 역 화일 방법보다 저장 공간의 효율이 좋다. 또한 레코드 요약 화일을 고분별력 단어에 대하여 결집하였기 때문에, 레코드 요약 화일의 접근을 위한 블럭 수가 감소되어 보다 나은 검색 성능을 달성할 수 있다.

III. 한글 텍스트를 위한 요약화일의 설계

본 논문은 먼저 한글 정보 검색 시스템 구축에 기반을 두고 이의 하부 구조로서 효율적인 검색 기법을 연구한다. 이를 위해 본 논문은 TITAN 시스템에서의 경험을 다각도로 분석하여 요약 화일 기법을 이용한 텍스트 검색 기법을 설계한다. 그러나 영문이 아닌 한글 텍스트인 점을 고려하여, 한글 텍스트를 위한 코딩 방법 (coding method), 한글 텍스트 코딩에 사용되는 해싱 기법, 어절의 분리 문제, 한글을 위한 요약 화일 구조에 중점을 두어 설계한다.

3.1 한글 텍스트 코딩 방법

한글 텍스트를 코딩하는 방법에는 크게 3 가지 접근 방법이 있다. 즉

(1) 한 음절을 해싱 단위로 코딩하는 방법
(2) 두 음절을 해싱 단위로 코딩하는 방법
(3) 한 음절과 두음절을 혼합하여 코딩하는 방법

처음 방법의 장점은 사용자에 의해 필요로 하는 모든 부분 매치 질의가 지원될 수 있다는 점이다. 그러나 한음절에 의한 코딩 때문에 한 어절을 검색하는 데 높은 false match 를 야기시키는 단점이 있다. 반면 두번째 방법은 평균 어절의 길이가 2.75 음절이라는 점을 감안하여 설계한 것으로 찾고자 하는 어절이 2 이상일 때는 매우 효과적인 방법이다 [15]. 그러나 한글에는 한 음절이 하나의 어절을 구성하는 경우가 많으므로 (예를 들면, 소, 말, 종, 길, 돌, 들, 산, 공, 배, 총, 손, 발, 비 등) 이를 검색할 수 없는 단점이 있다. 한편 두 개의 방법을 결합한 세번째 방법을 사용하면 위의 두 방법의 장점을 모두 지닐 수 있기 때문에 효율적인 코딩 방법을 설계할 수 있다.

일반적으로 영문 텍스트에서 요약 화일 기법을 사용하여 인덱싱하는 알고리즘은, 영문 텍스

트를 상수개의 단어를 (중복된 단어와 관사등 많이 쓰이는 공통어는 제외) 지닌 논리적 블럭으로 나눈 뒤, 블럭 내의 각 단어들을 해싱하여 각 단어들의 요약을 중첩 코드화하여 (superimposed coding) 인덱싱한다. 제안하는 세번째 방법을 이용, 한글 텍스트를 코딩하여 요약을 구성하는 알고리즘도 영문 텍스트와 마찬가지로, 주어진 텍스트를 논리적 블럭으로 나누어 각 블럭내에서 중첩 코드화 방법을 사용한다. 그러나 영문 텍스트와는 달리, 블럭내의 단어에 해당하는 어절 중심이 아닌 음절 중심의 해싱을 통한 요약을 구성하고 아울러 논리적 블럭을 상수개의 단어 즉 어절 중심이 아닌 상수개의 음절 중심으로 논리적 블럭을 구성하는 방법을 선택한다.

한편 영문 텍스트가 어절 중심인 반면 한글 텍스트가 음절 중심으로 나아가는 이유는, 첫째 조사등의 결합으로 인해 여백으로 분리된 부분이 순수 어절이기보다는 어절에 조사를 포함한 형태이므로 순수 어절을 분리하기 어렵고, 둘째 다수의 음절로 구성된 한 어절이 음절 사이의 여백을 허용하기도 하므로 여백에 의한 영문 텍스트 경우와 같은 어절 분리는 불가능하기 때문이다. 주어진 논리적 블럭이 B 이고 그에 따른 레코드 (논리 블럭) 요약 S 의 크기가 m 일때 한글 텍스트를 위한 요약 화일의 코딩 알고리즘은 다음과 같다. 한편 논리적 블럭은 음절 중심으로 구성되나, 나중에 사용자의 질의를 보다 용이하게 처리하기 위하여 한 어절과 다른 어절 사이에서 논리적 블럭은 경계를 이루며 분할된다.

<u>알고리즘</u>

(1) 주어진 논리적 블럭 B 에 포함된 각 음절에 대하여, 그의 해싱을 통해 주어진 요약 S 에서 0 에서 m-1 사이의 한 비트를 '1' 로 한다.

(2) 주어진 논리적 블럭 B 에 포함된 연속된 두 음절에 대하여 (N 개의 음절로 구성된 한 어절은 N-1 개의 연속된 두음절을 포함), 그의 해싱을 통해 주어진 요약 S 에서 0 에서 m-1 사이의 한 비트를 '1' 로 한다.

(3) 주어진 한글 텍스트가 n 개의 논리적 블럭으로 분할되었다면, 각 논리적 블럭을 통하여 구성한 요약 S1, ..., Sn 을 순차적으로 저장하여 주어진 한글 텍스트의 요약 화일을 구성한다.

제안된 방법의 효율성을 검토하기 위해 다음과 같은 가정하에서 평가한다. 먼저 논리적 블럭의 크기가 1K 바이트이고, 요약 화일의 크기가 원래 텍스트의 10% 인 점을 고려하여 요약의 크기 m 은 800 비트로 가정한다. 이때 1K 바이트의 한글 텍스트에서 발생하는 2 음절 패턴의 갯수는 평균 240 개, 중복을 제거한다면 160 개가 나타난다 [15]. 아울러 1 음절 수는 약 400 개, 중복을 제거하면 약 250 개 정도가 된다. 따라서 위의 알고리즘에 따라 요약 S 의 800 개의 비트에서 '1' 로 되는 비트수는, 중복을 제거한 1 음절과 2 음절 개수를 합한 값으로 약 400 개가 된다. 이 개수는 요약 화일 크기의 절반에 해당되는 값으로 이는 요약 화일 구성시에 false match 를 최소로 줄이는 최적의 요약 화일 구성으로 알려져 있다 [7]. 따라서 위와 같은 통계치에 의한 평가에 따르면, 제안된 세번째 코딩 방법은 사용자가 필요로 하는 모든 부분 매치 질의를 지원함은 물론 효율성 면에서도 아주 우수한 것으로 평가되고 있다.

3.2 해싱 기법

한글 텍스트의 두음절 패턴을 위해 기존에 3 가지 방법, 즉 모든 조합 가능한 두음절 패턴 개수와 동일한 길이를 갖는 비트 스트링을 축약하는 방법, 초성, 중성, 종성의 각 자모에 일련 번호를 붙여 해싱하는 방법, 그리고 음절의 빈도순으로 일련 번호를 붙여 비트를 분포시키는 방법 등이 제시되었다 [15]. 본 논문에서는 이와는 달리 영문 텍스트에서 ASCII 코드값에 따른 해싱 기법과 유사한 접근 방법을 택한다. 그 이유는 이러한 방법이 보다 간단하면서도 분산이 고르게 되는 해싱 함수를 만들기 때문이다. 요약의 크기가 m 이고 논리적 블럭 내의 한 음절이 c 일때, 그 음절을 0 에서 m-1 로 분산시키는 해싱 함수 f1 는 다음과 같다.

$$f1(c) = [\ p * \text{unsigned\_int}(c)\ ] \bmod m$$

여기서 unsigned_int() 함수는 한글의 한 음절 c 가 2 바이트 또는 3 바이트, 조합형 또는 완성형의 어떤 형태로 표현되었건 c 를 양의 정수로 변환하여 주는 함수이다. 아울러 p 는 30 보다 큰 솟수로 이의 곱셈을 통해 보다 균등한 분산을 달성할 수 있다. 아울러 요약의 크기가 m 이고 논리적 블럭 내의 연속된 두 음절이 각각 c1, c2 일때 해싱 함수 f2 는 다음과 같다.

$$f2(c1,c2) = [\ p1 * \text{unsigned\_int}(c1) + p2 * \text{unsigned\_int}(c2)\ ] \bmod m$$

여기서 p1, p2 는 30 보다 큰 서로 다른 솟수이며, 서로 다른 솟수를 택한 이유는 연속된 두 음절 패턴의 해싱 값과 그 두음절의 도치된 패턴의 해싱 값을 서로 다르게 만드기 위함이다. 즉 f2(c1,c2) 의 값과 f2(c2,c1) 의 값은 c1 과 c2 가 같지 않으면 서로 다른 값을 지닌다.

3.3 어절의 분리 문제

한글의 경우 한 어절이 하나의 의미를 전달하는 것은 틀림이 없지만, 영어의 구 (phrase) 와 같이 두개 또는 세개의 어절이 연결되어 하나의 새로운 어절을 구성할 때 이에 대한 처리 문제가 발생한다. 즉 '국민' '교육' '헌장' 의 3 개의 어절은 이들 고유한 뜻을 포함하고 있지만 한편 이들이 연결되어 '국민교육헌장' 이라는 하나의 새로운 어절을 구성한다. 이때 텍스트에는 같은 의미로서 '국민 교육 헌장' 이라고 저장이 되었을 때, 질의에 '교육헌장' 을 포함하는 텍스트를 찾고자 하면 이것이 질의를 만족하는 텍스트로 검색될 것인가 하는 문제가 발생한다. 분명 같은 의미이므로 검색되어야 한다면 상술한 코딩 방법은 개선 또는 변경이 되어야 한다. 왜냐하면 두 단어의 연결과 두 단어의 여백에 의한 분리는, 코딩 기법에 의해 요약을 구성하는 데 있어서는 완전히 다른 비트 패턴으로 나타나기 때문이다.

따라서 이를 처리하기 위해서 질의에서 나타난 어절을 의미 단위의 소 어절로 분리하여 처리하는 방법을 선택한다. 소 어절로의 분리를 질의 처리기가 자동으로 수행하여 처리할 경우,

사용자는 질의를 만들 때 소 어절로의 분리에 신경을 쓸 필요가 없는 장점이 있으나, 질의 처리기는 소 어절로의 분리를 위해 한글 디쩌러스를 구성하고 아울러 각 질의에 나타난 어절에 대해 한글 디쩌러스를 통해 소 어절로의 분리를 수행해야 하는 부담이 있다. 한편 한글 디쩌러스를 구성하는 것 자체도 한글 텍스트를 처리하기 위한 또 하나의 연구 과제이므로 본 논문에서는 한글 디쩌러스가 구성되어 있지 않다는 가정하에 질의 사용자에게 소 어절로의 분리를 맡기는 방법을 택하였다. 즉 위의 예에서와 같이 질의에 '교육헌장' 을 포함하는 텍스트를 찾고자 하면, 질의를 구성할 때 '교육 헌장' 과 같이 소 어절로의 분리를 사용자가 해 주어야 한다. 만약 소 어절로의 분리를 해주지 않으면 '국민교육헌장' 과 같이 여백없이 씌여진 텍스트만이 질의를 만족하는 텍스트로 검색된다. 한편 한글 디쩌러스의 구축이 다른 과제를 통해 이루어지면, 이를 질의 처리기에서 사용하여 사용자의 질의 구성시 소 어절로의 분리 부담을 덜어줄 수 있을 것이다.

3.4 요약 화일 구조

전술한 바와 같이 영문 텍스트를 위한 요약 화일 구조는 비트 슬라이스 요약 화일 구조, 2 단계 요약 화일 구조, 2 경로 2 단계 요약 화일 구조 등이 있다. 이들중 어떤 방법이 한글 텍스트를 위한 요약 화일 구조로서 적당할 것인지는 요약 화일 구성의 효율성 면에서 매우 중요한 문제이다. 한글 텍스트의 경우, 한글 텍스트의 코딩에 대해서는 언어 자체의 특수성 때문에 영문 텍스트와 상이한 방법이 선택될 수 있다. 그러나 일단 어떠한 코딩 방법을 통해 요약이 만들어지면 이 요약 화일의 구조에 대해서는 영문 텍스트의 경우와 다를 바가 없다. 따라서 기존 영문 텍스트를 위해 연구된 효율적인 요약 화일 구조가 그대로 적용되어 사용될 수 있다.

기존 영문 텍스트를 위해 연구된 요약 화일 구조에 대해 알려진 바에 따르면, 텍스트 화일 내의 텍스트를 포함하는 문서의 수가 많지 않을 시는 (문서의 수가 약 10 만개 이하 수준일 때) 비트 슬라이스 요약 화일 구조가 적합하며, 텍스트 화일 내의 문서의 수가 매우 많을 시는 (문서의 수가 약 100 만개 이상 수준일 때) 2 단계 요약 화일 구조가 적합한 것으로 알려져 있다. 그리고 데이타베이스의 키이의 개념과 같이 텍스트 내의 중요 단어에 (primary words) 대해 질의에서 빈번히 사용을 한다면, 즉 중요 단어를 질의에서 80% 빈도 이상으로 사용할 때, 2 경로 2 단계 요약 화일 구조가 타당한 것으로 알려져 있다. 한글 텍스트 경우에도 거의 유사한 원칙에 따라 적합성 여부가 적용될 것이며, 가장 주요한 선택상의 변수는 텍스트상의 문서의 개수와 텍스트 내의 중요 단어의 질의에서의 사용 빈도 등이다.

IV. 결 론

본 논문에서는 한국형 정보 검색 시스템 구축에 있어 그 하부 구조로서 필수적으로 요구되는 텍스트 검색 기법을 위해, 한글 텍스트의 특성을 분석, 파악하여 한글 텍스트에 적합한 요약 화일 기법을 설계하였다. 아울러 설계된 방법의 실용성과 효율성을 위해, 한글 텍스트를 위한 코딩 방법, 한글 텍스트 코딩에 사용되는 해싱 기법, 어절의 분리 문제, 한글을 위한 요약 화일 구조 등에 촛점을 맞추어 설계하였다. 앞으로 설계된 방법의 타당성을 보이기 위해 소규모의 한글 텍스트 검색 시스템을 구축하여 설계된 요약 화일 기법의 한글 텍스트와의 부합성을 검토하고 아울러 시스템 내에서의 한글 텍스트를 위한 검색 시간, 삽입 시간, 부가 저장 공간 등의 정량적인 평가도 수행할 예정이다.

참 고 문 헌

[1] V.J. Calderbank, "TITAN: An Information Management System for Faster Retrieval from Massive Database Using Signatures, Program, Vol. 24, No. 3, 1990, pp. 253-268.

[2] J.W. Chang et al., "Multikey Access Methods Based on Term Discrimination and Signature Clustering," Proc. of ACM SIGIR Conference, 1989, pp. 176-185.

[3] S. Christodoulakis and C. Faloutsos, "Design Considerations for a Message File Server," IEEE Trans. on Software Engineering, Vol. 10, No. 2, 1984, pp. 201-210.

[4] C. Faloutsos, "Access Methods for Text," ACM Computing Survey, Vol. 17, No. 1, 1985, pp.49-74.

[5] C. Faloutsos, "Signature Files: An Integrated Access Method for Text and Attributes, Suitable for Optical Disk Storage," BIT, Vol. 28, 1988, pp. 736-754.

[6] C. Faloutsos and S. Christodoulakis, "Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation," ACM Trans. on Office Information Systems, Vol. 2, No. 4, 1984, pp. 267-288.

[7] C. Faloutsos and S. Christodoulakis, "Design of a Signature File Method that Accounts for Non-Uniform Occurrence and Query Frequencies," Proc. of VLDB Conference, 1985, pp. 165-170.

[8] D.E. Knuth, "The Art of ComputerProgramming, Volume 3: Sorting and Searching," Addison-Wesley, 1973.

[9] L.A. Macleod, "A Model for Intergrated Information Systems," Proc. of VLDB conference, 1983, pp. 280-289.

[10] C.S. Roberts, "Partial-match Retrieval via the Method of Superimposed Codes," Proc. of IEEE, Vol. 67, No. 12, 1979, pp. 1624-1642.

[11] R. Sacks-Davis and K. Ramamohanarao, "A Two Level Superimposed Coding Scheme for Partial Match Retrieval," Information Systems, Vol. 8, No. 4, 1983, pp. 273-280.

[12] R. Sacks-Davis and K. Ramamohanarao, "Performance of a Multikey Access Method Based

on Descriptors and Superimposed Coding Techniques," Information Systems, Vol. 10, No. 4, 1985, pp. 391-403.

[13] R. Sacks-Davis et al., "Multikey Access Methods Based on Superimposed Coding Techniques, ACM Trans. on Database Systems, Vol. 12, No. 4, 1987, pp. 655-696.

[14] G. Salton and M.J. McGill, "Introduction to Modern Information Retrieval," McGraw-Hill Book Company, 1983.

[15] 송 병호, 이 석호, "요약 화일을 이용한 한글 사무문서 인덱싱 기법에 관한 고찰," '90 봄 학술발표 논문집, 한국 정보과학회, 1990, pp. 265-268.