대용량 한글 문서의 원문 보호 탐색 기법

(An Original Text Protecting Search Method for Huge Korean Documents)

박 선 영 † 김 성 환 †
(Sun-Young Park) (Sung-Hwan Kim)
조 환 규 ††
(Hwan-Gue Cho)

요 약 유사 문서 탐색 시스템의 개발이 꾸준히 이루어지고 있는 가운데, 유사 문서 탐색을 위한 데이터 수집 문제가 저작권과 관련하여 큰 문제가 되고 있다. 만약 유사 문서 탐색 시스템이 저작권자들의 저작물을 복원할 수 없도록 변환하여 보관하는 것을 보장한다면, 저작권자들이 데이터를 제공하는 데에 드는 거부감을 완화할 수 있을 것이다. 본 논문에서는 초성을 이용한 한글 스킨 추출 방법을 이용한 원문 보호가 이루어지면서 특정 단어나 문장이 존재하는지 탐색할 수 있는 시스템을 제안한다. 제안하는 시스템은 한글 문서의 초성을 추출하고, 버로우즈-휠러 변환(Burrows-Wheeler Transformation)을 수행하여 접미사 배열 정보와 원문 정보를 최소한의 용량으로 저장한다. 실험 결과 20자 이상의 문장에 대하여 신속하고 정확한 검색이 가능함을 보였다. 또한 1~2자의 불일치를 허용하는 탐색과 80% 부분 일치 탐색 방법을 제안하고 각각 5자, 15자 이상의 질의어에 대하여 효과적으로 동작함을 확인하였다.

· 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2011-0015359)
· 이 논문은 제38회 추계학술발표회에서 '원문 보호가 가능한 대용량 한글 문서 고속 탐색 기법'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 부산대학교 컴퓨터공학과
parksy@pusan.ac.kr
sunghwan@pusan.ac.kr
†† 종신회원 : 부산대학교 컴퓨터공학과 교수
hgcho@pusan.ac.kr
(Corresponding author임)

논문접수 : 2011년 12월 28일
심사완료 : 2012년 4월 9일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제18권 제7호(2012.7)

키워드 : 유사 문서 탐색, 버로우즈-휠러 변환, 초성 스킨

Abstract While similar document searching systems have been developed steadily, data collection to be searched is becoming a problem related to copyright law. If such searching systems guarantee that they store documents as a converted form which is secure and cannot be recovered, it makes much uncomplicated to get agreements from authors. In this paper, based on the fact that first phonemes of Korean sentences have much information than other phonemes, we propose a searching method which provides the protection of original text using Korean skin extraction. The proposed system extracts first phonemes(skin) from given Korean documents, and store them by BWT(Burrows-Wheeler Transform) to minimize the size while containing the information of original text and its suffix array. By experiment, we show that the searching is a quite fast and accurate with a query longer than 20. We also present a method to search allowing 1~2 mismatches and to find partial matching of 80%. We experimentally demonstrate these methods works effectively with queries longer than 5 and 15 respectively.

Key words : Similar Document Searching, Burrows-Wheeler Transform, First Phoneme Skin

1. 서 론

최근 연구 윤리의 중요성이 크게 부각되고 있으며, 그러한 현상의 하나로 유사 문서를 탐색하기 위한 많은 시스템이 개발되어 널리 활용되고 있다. 이러한 시스템은 기본적으로 사용자가 입력한 데이터 혹은 인터넷에서 수집한 데이터를 이용해 유사 문서 여부를 검사한다. 즉 사용자 수준에서 의심스러운 문서를 파일의 형태로 직접 가지고 있거나, 인터넷에서 언제든 가지고 올 수 있는 형태로 공개되어 있는 문서에 대해서만 검사가 가능하다. 하지만 대부분의 저작물은 인터넷에서 마음대로 받을 수 있는 성질의 것이 아니다. 대부분의 저작권자는 자신의 저작물이 인터넷에 공개되기를 원하지 않으며 이것은 표절 탐색을 위해서라도 마찬가지이다. 따라서 유사 문서 탐색 시스템에서 '데이터의 수집'은 아주 중요한 문제가 된다. 만약 유사 문서 탐색 시스템이 저장되어 있는 데이터의 보호를 보장할 수 있다면 저작권자들의 양해를 구하는 것이 조금 더 쉬울 것이다.

데이터의 유출을 방지하기 위하여 데이터베이스를 대칭키 또는 공개키 기반의 암호화 알고리즘을 이용하여 안전하게 암호화한 후 복호화를 하지 않고 검색을 할 수 있는 검색 가능 암호 기술에 대한 연구가 활발히 이루어지고 있다. 자세한 연구 동향은 [1]에 잘 소개되어 있다. 검색 가능 암호 기술은 사용자가 직접 질의어를 시스템

에 전송하는 것이 아니라 그에 대응하는 트랩도어를 생성하여 처리하는 방식을 통하여 질의어나 원문을 제3자에게 노출되도록 하지 않으면서 검색을 수행할 수 있다.

검색 가능 암호 기술은 안전한 검색 기법을 제공하지만, 역시 데이터의 수집 측면에서 저작물 전체에 대한 전적인 권리를 일임 받아야 한다는 점은 데이터베이스 구축에 여전히 장애물로 작용한다. 또한 키워드 기반의 색인을 전제로 하고 있으므로 표절 검사와 같이 문서간의 면밀한 비교 연산을 수행하기에는 적합하지 않다.

본 논문에서는 원문 보호가 가능한 초고속 한글 문서 탐색 시스템을 제안한다. 원문 보호를 위해 한글 스킨 추출 방법을 이용하고, 빠른 탐색과 저장을 위해 생물정보학에서 활용되는 압축 전처리 알고리즘인 버로우즈-휠러 변환(Burrows-Wheeler Transform)을 이용한다. 또한 질의어와 완전히 일치하는 구간에 대한 탐색과 함께 주어진 개수만큼의 불일치를 허용하는 탐색, 주어진 비율만큼의 일치 구간을 탐색하는 기법을 제안하고, 실험을 통해 성능을 검증하고자 한다.

2. 관련 연구

2.1 색인 방법

문서에서 특정 내용을 빠르게 찾기 위해 생성한 일련의 데이터를 색인(index)라고 한다. 사전 순으로 단어를 나열한 전통적인 찾아보기 방식의 색인이 가장 오래되고 널리 사용된 방법이나, 이 방법은 용량이 아주 크고 선형으로 탐색하므로 불리한 점이 많다. 이후 속도를 빠르게 하기 위하여 색인에 이진 탐색을 적용한 방법이 등장했고, 이후 해시 테이블을 이용한 색인 방법이 제안되어 널리 사용되다가 그 속도를 더욱 높인 접미사 트리와 접미사 트리보다 용량을 줄인 접미사 배열[2] 등이 활용되었다. 한편 버로우즈-휠러변환(Burrows-Wheeler Transformation, BWT)을 사용하면 본문과 동일한 크기의 공간에 접미사 배열과 본문의 내용을 동시에 저장할 수 있어, 이에 대한 활용 방법이 제안되고 있다[3].

2.2 초성 추출과 원문 보호 및 검색 성능

최근 연구에서 문자열에서 초성을 추출하여 원문을 보호하면서 탐색이 가능함을 보인 바 있다[4]. 이 방법은 한글 문자를 이루는 초중종성 중 초성의 정보량이 가장 높다는 점[5]을 이용하여 문장의 공백을 모두 제거하고 초성을 추출한 후, 쌍자음을 단자음으로 바꾸는 방법으로 초성 추출을 완성한다. 이를 한글 스킨 추출이라 하는데 길이 k 인 스킨으로부터 명사 사전을 이용하여 원문을 추측하기 위해서는 이론적으로 평균 45^k가지의 조합을 검토해 보아야 한다[4]. 한글 스킨 상에서의 탐색 성능에 대한 실험[4]에 따르면 문자열 길이 $k \geq 20$일 때, 60~99%가 일치하는 단어는 거의 존재하지 않으며, 100% 일치하는 경우가 1번 존재하거나 아예 존재하지 않는 것으로 양분됨을 밝혔다. 즉 20글자 이상으로 이루어진 문장은 우연히 겹치는 경우가 존재하지 않기 때문에 존재한다면 반드시 찾고, 존재하지 않는다면 반드시 존재하지 않음을 알 수 있다.

3. 한글 문서에 대한 보호 및 탐색 기법

2.2절의 초성 추출 방법을 사용해 한글 문서의 스킨을 추출하고, 이에 대한 접미사 배열을 생성한 후 BWT를 이용해 이를 저장함으로써 용량을 적게 사용하면서도 초고속 검색이 가능한 한글 문서 탐색 기법을 제안한다. 원문 보호가 가능한 동시에 대용량의 데이터에서 빠른 검색이 가능하다면 한글 스킨 데이터베이스를 구축함에 있어 각 저작자의 양해를 구하여 문서를 수집하는 것이 더욱 용이할 것이다.

3.1 초성 추출을 이용한 원문 보호

n개의 한글 글자로 이루어진 문서 $D=\langle x_1, \cdots, x_n \rangle$에 대하여 이 문서를 구성하는 각각의 한글 글자 x_i는 초성 f_i, 중성 m_i, 종성 l_i로 구성되어 있으므로 $x_i = \langle f_i, m_i, l_i \rangle$와 같이 표현할 수 있다. 한글 초성 변환 $Skin(D)$는 식 (1)과 같이 정의된다[4]. 또한 출현 빈도가 낮은 쌍자음 ㄲ,ㄸ,ㅃ,ㅆ,ㅉ 의 경우 출현빈도가 낮아 원문을 추측할 수 있는 가능성이 높아지므로 이를 단자음으로 변환하는 과정을 거친다.

$$Skin(D) = \langle cho(x_1), \cdots, cho(x_n) \rangle \quad (1)$$

$$cho(x_i) = \begin{cases} f_i & \text{if } f_i\text{가 단자음} \\ f_i\text{의 단자음} & \text{if } f_i\text{가 쌍자음} \end{cases} \quad (2)$$

예를 들어 x="깜"일 때 이는 <ㄲ,ㅏ,ㅁ>으로 표현되고 cho(깜)은 "ㄲ"의 단자음 "ㄱ"이 된다. 이와 같은 방법으로 문장 "이대호가 홈런을 때려내며 타격감을 뽐냈습니다."에 대하여 Skin 변환을 수행한 결과는 표 1과 같다.

표 1 한글 초성 변환의 예

D	이대호가홈런을때려내며타격감을뽐냈습니다
$Skin(D)$	ㅇㄷㅎㄱㅎㄹㅇㄷㄹㄴㅁㅌㄱㄱㅇㅂㄴㅅㄴㄷ

3.2 BWT를 이용한 한글 고속 탐색 방법

본 논문에서는 접미사 배열을 이용해 고속 탐색을 구현한다. 하지만 굳이 BWT 변환을 수행하는 이유는 특정 문서 집합에 대해 BWT 변환을 수행하면 BWT 변환된 시퀀스만으로도 원문과 접미사 배열을 매우 빠른 속도로 복원해 낼 수 있기 때문이다. 또한 원문에서 접미사 배열을 구하는 것보다 BWT에서 접미사 배열을

구하는 것이 빠르기 때문에, 데이터베이스를 구축할 때에 한글 스킨에 대한 BWT를 저장하는 것이 여러모로 유리하기 때문이다. BWT는 문자열에 종료 문자 '$'를 붙이고 한 글자씩 쉬프트한 후, 쉬프트된 각 문자열을 사전 순서대로 정렬한 결과의 마지막 글자를 순서대로 나열한 결과이다[3]. 그러나 시간 복잡도 문제로 BWT 정의를 이용하는 방법보다는 접미사 배열을 이용해 BWT를 생성하는 방법[6]를 사용하는데, 접미사 배열을 빠르게 생성하는 방법은 이미 많은 연구가 이루어져 있으므로 본 논문에서는 Kärkkäinen의 $\Theta(n)$의 복잡도를 가지는 알고리즘[7]을 사용하여 빠르게 접미사 배열을 구한 후 BWT를 수행하였다.

3.3 불일치를 허용하는 탐색 방법

먼저 1자의 불일치를 허용하는 탐색 방법을 설명한다. 그림 1과 같이 주어진 질의어 $Q=\langle q_1 \cdots q_n \rangle$이 있을 때, 이를 절반으로 절단하여 $Q_f=\langle q_1 \cdots q_{(n/2)} \rangle$와 $Q_l=\langle q_{(n/2)+1} \cdots q_n \rangle$을 구성한다. 만약 본문에 질의어와 정확히 1자만 일치하는 구간이 존재한다면 해당 구간은 반드시 Q_f와 Q_l 중 적어도 하나와 완전히 일치한다. 따라서 먼저 Q_f를 탐색하고 해당 위치에서 뒤로 Q의 n번째 문자까지 1번의 불일치를 허용하면서 확장하고, 같은 방식으로 Q_l을 탐색한 후 해당 위치로부터 앞으로 Q의 1번째 문자까지 확장하는 방식을 통해 1자 불일치를 탐색할 수 있다. 그림 2는 Q_f가 완전히 일치하고 불일치가 Q_l에 있는 경우에 대한 탐색 방법을 나타낸다. 이 때, FindExactMatching()은 접미사 배열을 기반으로 하여 문자열 S상에서 주어진 패턴이 완전히 일치하는 위치들의 집합을 반환하는 함수이다.

불일치 허용 횟수 k가 1보다 큰 경우 역시 질의어를 반으로 나누어 모든 $0 \leq i \leq k$에 대하여 절반에 대해서

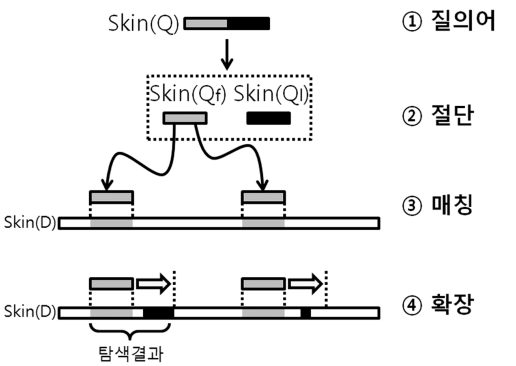

그림 1 불일치 1자를 허용하는 탐색 방법. 질의어를 절단하여 기준 문서에 매칭한 후 나머지 절반을 확장하여 탐색한다.

Algorithm Search_with_1Mismatch_in_LastHalf		
Input	S	탐색 대상 문자열
	P	탐색하려는 패턴
Output	$R=\{i\}$	패턴 P가 발견된 위치의 집합

```
Q_f ← P.substr( start=0, len=P.size()/2 )
for each f_i in FindExactMatching( S, Q_f )
  mismatch ← false
  for j = |P|/2 To |P|-1
    if( S[f_i+j] ≠ P[j] )
      if( mismatch ) next f_i
      else mismatch ← true
  R ← R ∪ {f_i}
return R
```

그림 2 불일치 1회를 허용하는 탐색 중 질의 패턴의 처음 절반이 일치하는 경우에 대한 탐색 알고리즘

는 i번의 불일치, 나머지 절반에 대해서는 $k-i$번의 불일치를 허용하도록 재귀적으로 탐색하는 방법을 통하여 구현할 수 있다. 다만 이 경우 k가 증가함에 따라 조합해야 하는 경우의 수에 비례하여 탐색시간이 급격하게 증가하므로 k가 충분히 작을 경우에만 효율적으로 동작할 수 있다.

3.4 부분 일치 탐색 방법

불일치 허용 횟수 k가 증가하는 경우에는 일치 비율 r에 따른 부분 일치 탐색을 수행하도록 한다. 예를 들어 Q가 "반갑습니다"의 초성인 "ㅂㄱㅅㄴㄷ"이고 $r=0.5$인 경우 "ㄱㅊㅅㄴㄷ"나 "ㅂㅅㅎㄴㄷ" 등 $\lceil 5\times0.5 \rceil$=3글자 이상 일치하는 문자열을 탐색할 수 있어야 한다. 부분 일치 탐색은 그림 3과 같은 과정을 통해 수행된다. 먼저 질의어를 복사하여 10여개의 복사본을 만든 후, 이들을 임의의 길이로 절단한다. 절단된 질의어의 짧은 구간들을 본문에서 탐색한 후 특정 구간에서 부분 질의어가

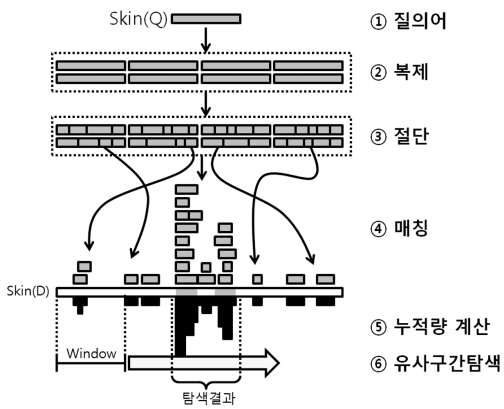

그림 3 일정 비율 부분 일치 허용 탐색 기법. 질의어를 복제하여 임의의 길이로 절단한 후, 절단 부분들이 일정 비율 이상 누적된 구간을 탐색한다.

Algorithm Search_PartialMatching

Input S 탐색 대상 문자열
P 탐색하려는 패턴
k 패턴 복제 횟수
r 일치 비율

Output $R=\{i\}$ 패턴 P가 발견된 위치의 집합

$\{p_i\} \leftarrow$ Copy_and_Cut(P, k)
$H \leftarrow$ new vector($|S|$)
for each p_i in $\{p_i\}$
  for each l_i in FindExactMatching(S, p_i)
    $H[l_i] \leftarrow H[l_i] + |p_i|$
$R \leftarrow \left\{ i : \sum_{j=i}^{i+|P|-1} H[j] \geq r \cdot k \cdot |P| \right\}$
return R

그림 4 부분 일치 탐색을 위한 알고리즘

반복해서 등장하는 경우 일치 비율을 계산하여 실제 질의어와의 일치율을 비교하여 기준 비율 이상인 구간을 찾아내어 유사구간으로 판단한다.

그림 4는 부분 일치 탐색 방법에 대한 의사코드를 나타낸다. 이 때, Copy_and_Cut()은 문자열을 입력받아 복제하고 절단한 문자열들의 집합을 반환하는 함수이다. FindExactMatching()은 이전 절에서 설명한 바와 같이 주어진 질의 패턴이 완전히 일치하는 위치를 접미사 배열 상에서 탐색하는 함수이다.

4. 시스템 성능 평가를 위한 실험

초성 탐색을 이용한 원문 보호 방법의 보호 성능에 대해서는 이미 입증이 되어 있으므로 본 논문에서는 BWT 생성 속도와 탐색 속도를 중심으로 실험을 진행하였다. 실험을 위한 컴퓨팅 환경은 표 2와 같다. 실험을 위해 영문 및 한글이 포함된 세종 말뭉치 데이터[8]를 표 3과 같이 용량 별로 잘라서 실험 데이터를 생성하였다. 생성된 실험 데이터에 대한 BWT 생성 시간을 측정하고, 한글 문자열에 대한 탐색을 수행하였다.

BWT 생성에 관한 실험 결과는 표 4와 같다. 전체 BWT 생성 시간의 복잡도는 $O(n)$을 따르며, 이 중 접미사 배열의 생성에 소요되는 시간이 가장 긴 것을 확인할 수 있다.

완전 일치 검색의 실험 결과는 표 5와 같다. 실험 결과를 살펴보면 검색 결과의 개수와 문자열의 길이에 상관없이 0.01초 이내로 검색이 수행되어 실시간 검색이 가능함을 알 수 있다. 표 6은 1자 및 2자의 불일치와 80% 부분 일치를 허용한 탐색 결과를 나타낸다. 검색 결과 수가 지나치게 많거나 탐색시간이 긴 경우는 실제로 활용되기 어려우므로 결과에서 제외하였다.

표 2 실험을 위한 컴퓨팅 환경

항목	상세
CPU	Intel Core i5 750 (2.67GHz×4)
RAM	Samsung DDR3 4GB
HDD	Intel SSD 80GB (Read 250MB/s, Write 70MB/s)

표 3 실험을 위해 '21세기 세종 말뭉치[8]'를 일정 비율로 취합, 절단하여 생성한 실험 데이터

번호	크기(KB)	어절 수	음절 수
1	10	1,519	6,106
2	80	10,442	50,607
3	400	53,505	407,779
4	1,000	153,420	1,034,869
5	7,000	1,020,117	7,175,185
6	63,000	9,223,169	65,297,029
7	101,000	14,280,303	104,193,793

표 4 각 데이터 별 BWT 생성 시간. 접미사 배열 생성 시간이 대부분을 차지한다.

번호	스킨 추출 (s)	접미사배열 생성 (s)	BWT생성 (s)	합계 (s)
1	0.01	0.01	0.01	0.03
2	0.01	0.04	0.01	0.06
3	0.01	0.18	0.01	0.20
4	0.07	0.45	0.01	0.53
5	0.50	5.41	0.09	6.00
6	4.54	71.54	0.99	77.07
7	6.92	127.66	1.67	136.25

표 5 문자열 검색 결과 실험. 검색 결과의 개수와 단어의 길이에 상관없이 검색 시간은 모두 0.01초로, 사실상 실시간 검색이 가능한 것을 확인할 수 있다.

번호	Q길이	정답 개수	결과 개수	검색 시간(s)
1	2	212	329,241	0.01
2	5	43	50	0.01
3	6	7	8	0.01
4	5	16	34	0.01
5	4	19	19	0.01
6	4	6	6	0.01
7	5	1	3,087	0.01
8	15	1	1	0.01
9	132	1	1	0.01
10	323	1	1	0.01
11	684	1	1	0.01

5. 결 론

본 논문에서는 한글 초성을 이용한 스킨 추출 방법과 BWT를 활용하여 한글 문서에 대한 보호를 유지하면서 문장 혹은 단어를 탐색하는 시스템을 제안하고, 그 성능을 실험을 통해 검증하였다. 결론은 다음과 같이 정리할 수 있다.

표 6 불일치 허용 및 부분 일치 검색 실험. 결과가 2백만개 이상으로 지나치게 많거나 탐색에 100초 이상이 소요되어 실제로 사용하기 힘든 경우는 결과에서 제외하였다. 1자 불일치 허용은 최소 5자, 2자 불일치 허용과 80% 부분 일치 검색에서는 15자 이상의 질의어인 경우 사용 가능함을 확인할 수 있다.

번호	Q 길이	정답 개수	불일치 허용(1)		불일치 허용(2)		부분 일치(80%)	
			결과 개수	검색 시간(s)	결과 개수	검색 시간(s)	결과 개수	검색 시간(s)
1	2	212	-	-	-	-	-	-
2	5	43	69,299	2.35	1,688,993	82.42	-	-
3	6	7	2,867	0.01	298,026	32.17	-	-
4	5	16	17,861	0.76	714,332	52.32	-	-
5	4	19	4,074	12.34	-	-	-	-
6	4	6	117	0.04	1,818	0.22	-	-
7	5	1	12	0.01	3,007	0.13	-	-
8	15	1	2	0.01	2	0.01	2	7.12
9	132	1	1	0.01	1	0.01	1	1.37
10	323	1	1	0.01	1	0.01	1	0.94
11	684	1	1	0.01	1	0.01	1	0.63

1) 한글 초성 문자에서 원문을 복원할 확률은 사전을 사용하더라도 k글자의 단어에 대해 45^{-k}에 수렴하므로, 실제로 원문을 추적해 낼 확률은 0에 가깝다.
2) 원문을 추적할 수는 없으나, 반대로 20글자 이상의 동일한 문장이 있다면 우연히 일치할 확률이 0에 가까워지므로, 같은 문장은 반드시 찾아낼 수 있다.
3) BWT 변환을 위한 시간은 순수 텍스트 기준 100 MB 크기의 문서를 기준으로 3분 이내이다.
4) 완전 일치 검색 시간은 문장의 길이와 동일한 문장의 개수와 관계없이 0.01초 이내에 이루어진다.
5) 1자의 불일치를 허용하는 검색의 경우 5자 이상의 질의어에 대하여 효과적으로 동작하며 15자 이상의 질의어에 대하여 0.01초 내에 이루어진다.
6) 2자의 불일치 허용 검색과 부분 일치(80%) 검색의 경우 15자 이상의 질의어에 대하여 효과적으로 동작한다.

실험 결과에 따르면 초성을 이용하여 한글 문서의 원문을 강력하게 보호하는 동시에 빠른 탐색을 수행할 수 있기 때문에 제안하는 시스템은 상당히 강력한 도구가 될 수 있다. 다만 현재는 검색 기법에 대한 제안과 결과론적인 효과성의 검증에 그친 수준으로 각 탐색 기법에 대한 최적화와 검색 성능 평가, 그리고 다양한 활용 기법에 관한 보다 심도 있는 연구가 향후에 진행되어야 할 것이며, 한글 이외의 문자로 이루어진 문서에 적용할 수 있는 방안을 모색해야 할 것이다.

참 고 문 헌

[1] 김선영, 서재우, 이필중, “검색 가능 암호 기술의 연구 동향”, *정보보호학회지*, 제19권 제2호, pp.73-90, 2010.
[2] U. Manber and G. Myers, "Suffix arrays : a new method for on-line string searches," *SIAM Journal on Computing*, vol.22, no.5, pp.935-948, 1993.
[3] M. Burrows and D. Wheeler, "A block sorting lossless data compression algorithm," *Tech. Rep., Technical Report 124, Digital Equipment Corporation*, 1994.
[4] 김성환, 박선영, 조환규, “한글 초성을 이용한 원문보호 탐색기법”, *한국정보처리학회 춘계학술발표대회 논문집*, 제18권 제1호, pp.396-389, 2011.
[5] 이재홍, 오상현, “한글 음절의 초성, 중성, 종성 단위의 발생확률, 엔트로피 및 평균상호정보량”, *전자공학회논문지*, 제27권 제9호, pp.1299-1307, 1989.
[6] G. Lee, K. Park, "Efficient Storing of Suffix Arrays using Block - Sorting Compression," *Journal of KIISE : Computer Systems and Theory*, vol.28, no.7·8, pp.350-355, Aug. 2001. (in Korean)
[7] J. Kärkkäinen and P. Sanders, "Simple linear work suffix array construction," *in ICALP*, pp.943-955, 2003.
[8] 21세기 세종계획, http://www.sejong.or.kr/, 2011.