PDA를 이용한 한국어 자동 색인 시스템

Korean Automatic Indexing System Using the PDA

박평구, 정인정
고려대학교 전산학과

Pyeung-Koo Park, In-Jeong Chung
Dept. of Computer Science, Korea Univ.

요 약 인터넷의 급속한 발달로 사용자는 자신의 질의에 적합한 검색결과를 빠르고 정확하게 보장하는 검색도구를 요구하게 되었다. 이러한 사용자의 요구는 검색도구의 성능향상에 필수적인 문서의 내용을 대표하는 색인어를 추출하는 색인 시스템에 대한 관심을 가지게 되었다. 기존의 한국어에서의 자동 색인 방법에는 어절 중심 색인법, 형태소 중심 색인법과 최근에 n-gram 중심 색인법등이 주류를 이루어 왔다. 그러나 한국어에서 색인어를 추출하는 기존의 방법은 복합명사의 색인과 복잡한 문법적 지식이 필요하고 잘못된 색인어를 추출하는 등의 검색효율에 문제점을 가지고 있다. 본 논문에서는 PDA를 이용한 정형화된 한국어와 영어문장의 자동 색인 방법을 제안한다. 제안하는 방법은 별도의 사전지식이 필요하지 않고 단일명사와 복합명사의 색인이 가능하며 인터넷으로의 확장과 다양한 언어로 확장성이 좋은 장점 등을 갖는다. 성능 평가로써 한국통신의 KTSET으로 MS사의 IIS를 웹 서버로 ASP를 이용하여 인터넷 환경에서 테스트를 통하여 한국어 뿐 아니라 영어문장의 정형화되고 이용이 간편한 자동색인 결과를 보여준다.

1. 서론

인터넷의 보급으로 방대한 문서들의 집합에서 사용자의 질의에 정확하고 빠른 응답에 대한 요구가 증가하게 되었다. 이러한 요구는 사용자의 질의에 대한 정확하고 빠른 응답을 보장하기 위해서 문서들의 특성을 가장 잘 반영하는 색인어(index term)를 선택하는 색인 시스템(index system)에 관심을 가지게 되었다[1].

그동안 한국어의 색인 방법은 어절 중심 색인법(word based indexing)[2,3]과 형태소 중심 색인법(morpheme based indexing)[3,5], 최근에 n-gram 중심 색인법(n-gram based indexing)[2,3,4]등이 주류를 이루어 왔다.

어절 중심 색인법은 단어에서 비색인 어절을 분리하고 남은 어절을 색인어로 추출한다. 그러나 한국어와 같이 복합명사가 많이 포함된 언어의 색인에는 효율적인 색인을 추출할 수 없다[2,3]. 또한, 형태소 중심 색인법은 단어를 구성하고 있는 형태소들을 인식하여 대부분의 언어에서 공통적으로 발생하는 언어현상인 형태론적 변형을 처리하는 기법을 기반으로 한다[5]. 이 색인법 역시 복합명사와 합성명사의 처리는 가능하지만 복잡한 언어특성을 포함하는 사전(dictionary)을 구성해야하고 언어의 문법구조를 이해해야 하는 문제를 포함하고 있다. 또한 사전의 참조로 인해 검색의 응답 시간의 저하를 가져온다[5]. 마지막으로 최근의 관심연구인 n-gram 중심 색인법은 어절을 n (n = 1,2...m , m=단어의 길이)에 의해서 분절함으로써 색인어를 추출하는 방법이다. 이 색인법은 복잡하지 않고 복합어의 분리 등에 유용하다[3,4,6]. 그러나 n 값의 결정방법에 따라 잘못된 색인어 추출로 색인어 추출효율 차이가 심한 단점이 있다.

본 논문은 기존의 한국어 자동 색인법의 장점을 보장하고 확장성이 뛰어나며 복합명사뿐만 아니라 새로 추가되는 신조어와 중의적 분해가 가능하도록 PDA(push down automata)를 이용한 정형화된 한국어 자동 색인법을 제안한다. 제안하는 방법은 인터넷과 다양한 언어로 확장이 가능하다. 성능평가에서 실제 인터넷상의 문서화된 KTSET을 대상으로 단일명사와 복합명사의 색인성능을 평가하고 영어문장을 제안하는 시스템에 적용하여 다른 언어로의 확장성과 음절수에 관계없이 색인어를성공적으로 추출하는 것을 보여준다. 제안하는 시스템은 별도의 사전지식이 필요하지 않고 정형화된 기계를 통해서 한국어 뿐 아니라 다른 언어로의 확장이 가능한 방법이다.

2. 오토마타(Automata)와 PDA(Push Down Automata)

본 논문은 PDA를 이용하여 한국어로 구성된 문서의 자동색인 방법을 제안하고 있다. 다음은 제안하는 색인방법에서 PDA를 이해하기 위한 계산이론 내용이다.

PDA는 촘스키 계층구조(Chomsky Hierarchy)의 형식언어(formal language)들에서, 문맥 자유 문법(context free grammar)에 의해 생성되는 형식 2언어인 문맥 자유 언어(context free language)를 수락한다[10].

PDA 기계 M의 정의는 $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$로 표현된다. 여기서 Q는 상태들의 유한 집합, Σ는 입력 알파베트(input alphabet), Γ는 스택 알파베트, $q_0 \in Q$는 최초 상태(initial state), $z_0 \in \Gamma$는 스택의 시작 부호(start symbol), $F \subseteq Q$는 최종상태(final state)들의 집합이다. 마지막으로, δ는 $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$에서 $Q \times \Gamma^*$의 유한 부분 집합으로의 사상(mapping)이다.

PDA는 입력 테이프, 유한 제어(finite control), 그리고 저장 공간으로 스택을 갖고 있다. 아래의 그림 1는 PDA의 일반적 구성이다[9].

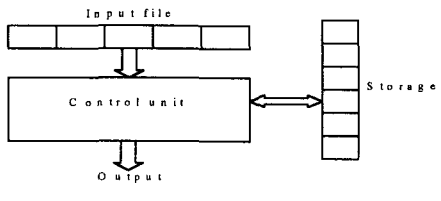

그림 1. PDA의 구성

우리가 본 논문에서 다루는 PDA는 상황에 따라 2가지 이상의 동작들 중에서 어떤 옳은 선택을 할 수 있다는 점과
입력 부호가 사용되지 않고도 동작을 취할 수 있는 ε-동작이 허용되는 점에서 비결정적(non-deterministic)인 PDA이다. 한편, 문맥 자유 언어가 PDA에 의해 수락될 때 2가

지 PDA에 의해 수락된다. 첫 번째 PDA는 빈 스택(empty stack)에 의해 문맥 자유 언어를 수락하는 방법이며, 두 번째 PDA는 스택의 내용과 상관없이 수락 상태(accepting state)들에 의해 문맥 자유 언어를 수락하는 방법이다. 만약, 어떤 언어가 빈 스택의 PDA에 의해 수락되면, 이 언어는 수락 상태에 의한 다른 PDA에 의해서도 수락되며, 그 역도 성립한다.

3. PDA를 이용한 한국어 자동색인 방법

위에서 언급한 일반적인 PDA를 한국어의 자동색인 시스템에 적용하기 위해서 입력되는 문장을 제어기가 읽어 미리 정의 된 PDA에 의해서 다음의 상태를 결정하고 스택에 읽은 문자를 저장하는 반복된 과정을 수행하게 된다. 이 때 수락단계에서 스택에 저장되었던 문장에서 색인어로 추출(pop)한다. 다음의 알고리즘은 의사코드(pseudo-code)의 형태로 설명하고 있다.

3.1 알고리즘

본 논문에서 제안하는 자동 색인 방법은 크게 3개의 단계로 이루어진다. 첫 번째 단계는 문서내의 문장을 인식하는 단계이고, 두 번째 단계는 인식된 문장에서 불용어 리스트(stop-list)를 임의의 문자로 치환하는 단계이다. 본 논문에서 사용된 방법은 문장 내에서 색인어로 부적절한 어절의 접두어와 접미어를 포함한 형용사와 동사의 용언을 뜻한다. 한국어에서 색인어로 추출이 가능한 어절은 대부분이 명사이기 때문이다. 마지막 단계는 PDA를 이용한 색인어 추출 단계이다. 아래의 의사코드는 마지막 단계인 색인어를 추출하는 알고리즘이다. 아래의 알고리즘에서 스택 심볼을 찾는 모듈과 상태를 찾는 모듈은 다음절의 PDA를 통한 구현을 통해서 알아본다.

```
input : 불용어가 치환된 문장
output : 색인어 리스트
char ch;
FILE *file_pointer
while (EOF)   /* 파일의 끝까지 색인방법을 적용한다. */
{   while(EOL)
    {   ch=getchar(file_pointer);
        if (ch==character)
            push_stack();
/* 입력된 문자가 공백이거나 불용어가 아니면 스택에 문자를 push 한다. */  else
            pop_to_buffer();
    }
}
```

위의 메인 모듈은 파일에서 문장단위로 불용어가 제거된 입력테이프에서 읽은 입력이 문자이면 문자를 스택에 push하는 모듈을 아니면 색인어를 스택에서 추출하는 모듈을 호출하게 된다.

```
/* push 모듈 */
push_stack()
{   stack[top] = ch;
    stack_symbol[top] = find_stack_symbol();
    stack_state[top] = find_state();
/* 입력된 문자를 스택에 push하고 스택 심볼을 찾아 push 한다. */
}
```

push_stack() 모듈은 읽은 문자를 스택의 top에 push하고 스택 심볼을 찾는 모듈을 통해서 스택의 심볼을 찾아 스택에 push한다. 또한 상태(state) 정보를 찾아 각각 top의 위치에 push를 수행한다. *find_stack_symbol*()는 현재 스택의 스택 심볼과 상태로 PDA를 바탕으로 스택 심볼을 결정하는 모듈이다. *find_state*() 역시 스택의 top 심볼과 현재 스택 top의 상태 입력된 문자를 바탕으로 상태를 결정하는 모듈이다. *find_stack_symbol*()과 *find_state*() 모듈은 다음의 PDA에서 설명하도록 한다.

```
/* pop 모듈 */
pop_to_buffer()
{   if (ch==stop_list||ch==space)
    {   stack_state[top]=2;
        stack_symbol[top]=G;
/* 스택 심볼을 G로 어절 단위로 나누어 스택에 저장하게 된다. */
        while (stack_symbol!=G)
        {
/* 이전의 어절까지 스택에서 pop하여 색인어로 추출한다. 즉, 복합명사 색인을 위한 것이다.*/
            buffer[]=stack[top--];
            index_term=buffer[];
            stack_state[top]=3;
        }
        pop_to_bottom();
/* 복합 명사를 색인어로 추출하기 위하여 스택의 바닥 심볼까지 pop한다. */
    }
}
```

pop을 수행하는 모듈에서 우리는 PDA가 비 결정적이라는 것을 알 수 있다. 그러므로 스택의 top에서 스택 심볼이 G일 때까지 문자를 pop해서 색인어로 추출하고 상태를 3으로 변환한다. 스택의 심볼에 G인 경우는 공백이거나 불용어를 뜻하므로 어절단위로 공백이나 불용어를 분리하여 스택에 push 또는 pop한다.

```
pop_to_bottom()
{   /* 모든 스택의 문자를 pop해서 복합명사를 색인어로 추출 */
    while (stack_symbol!=G)
    {   buffer[]=stack[top--];
        index_term=buffer[];
    }
}
```

pop_to_bottom()는 부분적인 색인어를 추출하였으므로 복합명사의 색인을 위해 이미 저장되어 있는 스택의 모든 심벌을 추출하여 색인어로 추출한다.

3.2 PDA 설계

위의 PDA를 이용한 한국어 자동색인 방법의 알고리즘에서 PDA는 다음과 같이 정의 할 수 있다. $M=(\{q_1, q_2, q_3\}, \{character, space, stop_list\}, \{R, B, G\}, \delta, \{q_1, q_3\}, R, \varnothing)$은 그림3에 정의한다. 정의된 PDA는 상태 q_1, q_2, q_3로 구성되어 있고, 입력은 문자, 공백, 불용어 리스트로 구성된다. 또한 스택 심벌은 처음 스택을 의미하는 R과 공백이나 불용어를 의미하는 G, 문자의 입력을 의미하는 B로 구성된다. q_1과 q_3를 최종 수락 상태로 한다. 다음의 그림 2은 위의 정의된 PDA를 나타낸다.

$$\delta(q_1, character, R) = \{(q_1, BR)\}$$
$$\delta(q_1, character, B) = \{(q_1, BB)\}$$
$$\delta(q_2, character, G) = \{(q_1, BG), (q_2, \varepsilon)\}$$
$$\delta(q_1, space, B) = \{(q_2, BG), (q_2, \varepsilon)\}$$
$$\delta(q_2, space, G) = \{(q_2, G)\}$$

$\delta(q_1, space, R) = \{(q_1, R)\}$
$\delta(q_1, stop-list, R) = \{(q_1, R)\}$
$\delta(q_1, stop-list, G) = \{(q_2, G)\}$
$\delta(q_1, stop-list, B) = \{(q_2, GB), (q_2, \varepsilon)\}$
$\delta(q_2, \varepsilon, R) = \{(q_1, \varepsilon)\}$
$\delta(q_2, \varepsilon, G) = \{(q_2, \varepsilon), (q_3, \varepsilon)\}$
$\delta(q_2, \varepsilon, B) = \{(q_2, \varepsilon)\}$
$\delta(q_1, EOL, B) = \{(q_2, GB)\}$
$\delta(q_2, EOL, G) = \{(q_2, G)\}$

그림 2. PDA를 이용한 한국어 자동 색인 방법

제안하는 PDA를 이용한 한국어 자동 색인 방법에서 시작 상태는 q_1이고 수락하는 상태는 q_1과 q_3 이다. 초기 상태가 에서 $\delta(q_1, character, R)$는 상태 q_1이고 스택 심볼이 R일 때 입력테이프의 입력이 문자(*character*)이면, $\{(q_1, BR)\}$와 같이 상태는 그대로 1이 되고 스택 심볼만 BR로 치환된다. 이때 입력된 문자는 스택에 push된다. 그런데 $\delta(q_1, space, B)$는 상태 1이고 스택 심볼이 B이고 입력 테이프가 공백 일 때 이미 앞에 언급된 대로 본 논문에서 제안하는 PDA는 비 결정적이므로 $\{(q_1, GB)\}$인 경우와 $\{(q_2, \varepsilon)\}$로 상태가 변환된다. 또한 $\delta(q_2, \varepsilon, R)$의 의미는 상태2에서 입력의 상태에 관계없이 스택 심볼이 R인 경우는 상태 1에서 수락하는 최종단계를 의미하게 된다. $\delta(q_2, \varepsilon, G)$는 입력 문자에 관계없이 스택 심볼이 G이면 상태는 2로 되면서 동시에 문자 하나 pop하고 다른 경우는 q_3상태로 되고 문자하나를 pop 하면서 최종 수락 상태이므로 이미 pop 되었던 문자를 색인어로 선택하게 된다.

3.3 예제

예제는 위의 정의된 PDA를 이용하고 한국통신에서 정보검색 도구의 성능을 평가하기 위해 개발된 테스트 데이터의 모음인 KTSET을 이용하기로 한다. 문서에서 문장을 인식하는 제 1 단계에서 입력 테이프에 입력된 문장이 "컴퓨터 바이러스나 백신에 대한 문서"라 가정한다. 제 2 단계의 불용어 리스트를 임의의 문자로 치환하는 단계를 통해서 "컴퓨터 바이러스^ 백신^ ^ 문서"가 이루어진다. 예제에서 편의상 "^"는 불용어를 의미하고 공백은 □으로 표시한다. 제 3 단계를 설명하기 위해서 PDA의 순간묘사(instantaneous description : ID)를 이용한다. 그림 3는 예제에 대해서 PDA를 통한 색인 과정을 보인다. *EOL*(end of line)은 문장의 끝을 나타낸다.

$\delta(q_1, 컴퓨터\,\_바이러스\,\hat{}\,\square백신\,\hat{}\,[\,]\,\hat{}\,\_문서, R)$ ①
$\vdash\delta(q_1, 퓨터\,\_바이러스\,\hat{}\,\square백신\,\hat{}\,[\,]\,\hat{}\,[\,]문서, BR)$
$\vdash\delta(q_1, 터\,\_바이러스\,\hat{}\,\square백신\,\hat{}\,\_\,\hat{}\,\_문서, BBR)$
$\vdash\delta(q_1, \_바이러스\,\hat{}\,\square백신\,\hat{}\,\_\,\hat{}\,\square문서, BBBR)$ ②
$\vdash\delta(q_2, 바이러스\,\hat{}\,\square백신\,\hat{}\,\_\,\hat{}\,\square문서, GBBBR)$ ③
$\vdash\delta(q_1, 이러스\,\hat{}\,\square백신\,\hat{}\,\_\,\hat{}\,\_문서, BGBBBR)$ ④

그림 3. PDA를 통한 색인 방법 예제

위의 단계 ①은 처음 상태 1에서 스택 심볼이 R인 경우 '컴'이라는 문자를 읽었으므로 앞 절에서 정의된 PDA에 의해서 상태는 계속 1이 되고 스택 심볼은 BR로 치환이 이루어진다. 단계 ②에서 변환될 때 제안하는 PDA는 비결정적이므로 단계 ③과 단계 ④로 동시에 변환이 가능하게 된다. 단계 ②에서 상태 1이고 스택 심볼이 B이면 공백을 읽으면 상태는 2가 되고 스택 심볼에 G가 삽입되는 단계 ③으로 변환된다. 또는 단계 ②에 같은 조건에서 계속해서 입력 테이프를 읽는 변환은 단계 ④를 의미한다. 이때 알고리즘을 통해서 알고 있듯이 단계 ③이후의 변환과정은 어절 단위의 색인을 위한 것이고, 단계 ④는 복합명사의 색인을 위해 계속해서 입력을 받는 과정이다. 단계 ③에 pop을 수행하여 색인어를 추출하는 변환이 그림 4에 나타내고 있다.

$\vdash\delta(q_2, 바이러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, GBBBR)$ ③
$\vdash\delta(q_2, 바이러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, BBBR)$
$\vdash\delta(q_2, 바이러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, BBR)$
$\vdash\delta(q_2, 바이러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, BR)$
$\vdash\delta(q_2, 바이러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, R)$
$\vdash\delta(q_1, 바이러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, \varepsilon)$ ⑤

그림 4. PDA를 통한 색인 방법 예제

위의 그림의 단계 ⑤까지 스택에서 "컴퓨터"라는 어절을 pop 하여 색인어로 추출하는 과정을 나타내고 있다. 이때 3.2절의 PDA에서 알 수 있듯, 입력테이프에 남아있는 문자에 관계없이 상태 2에서 스택 심볼이 모두 pop 될 때까지 반복된다. 그림 5는 그림 3의 단계 ④에서 계속해서 색인 작업을 위한 스택에 push가 반복적으로 이루어지는 것을 나타낸다.

$\vdash\delta(q_1, 이러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\_문서, BGBBBR)$ ④
$\vdash\delta(q_1, 러스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, BBGBBBR)$
$\vdash\delta(q_1, 스\,\hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, BBBGBBBR)$
$\vdash\delta(q_1, \hat{}\,\square백신\,\hat{}\,\square\,\hat{}\,\square문서, BBBBGBBBR)$ ⑥
$\vdash\delta(q_2, \square백신\,\hat{}\,\square\,\hat{}\,\square문서, GBBBBGBBBR)$ ⑦
$\vdash\delta(q_2, 백신\,\hat{}\,\square\,\hat{}\,\square문서, GBBBBGBBBR)$ ⑧

그림 5. PDA를 통한 색인 방법 예제

앞 절에 제시된 PDA대로 단계 ④에서 상태 1이고 스택의 심볼이 B이고 입력된 문자가 불용어이면 단계 ⑥까지 반복해서 스택에 push 과정이 수행된다. 단계 ⑦에서는 스택 심볼이 이미 G이므로 변환이 발생하지 않고 스택의 pop 단계를 수행한다. 아직 문장의 끝을 나타내는 *EOL*을 읽지 못했기 때문에 계속해서 PDA 색인 작업을 수행한다. 여기서는 단계 ⑧에서 pop 과정은 "컴퓨터 바이러스"라는 복합어가 색인어로 추출되는 것을 볼 수 있다. 단계 ⑧의 복합어 색인을 위한 pop 과정은 그림 6에서 나타내고 있다.

$\vdash\delta(q_2, 백신\,\hat{}\,\square\,\hat{}\,\square문서, GBBBBGBBBR)$ ⑧
$\vdash\delta(q_1, 백신\,\hat{}\,[\,]\,\hat{}\,[\,]문서, BBBBGBBBR)$
$\vdash\delta(q_2, 백신\,\hat{}\,[\,]\,\hat{}\,[\,]문서, BBBGBBBR)$
$\vdash\delta(q_2, 백신\,\hat{}\,[\,]\,\hat{}\,\square문서, BBGBBBR)$
$\vdash\delta(q_2, 백신\,\hat{}\,\square\,\hat{}\,\square문서, BGBBBR)$
$\vdash\delta(q_3, 백신\,\hat{}\,\square\,\hat{}\,\square문서, GBBBR)$ ⑨
$\vdash\delta(q_3, 백신\,\hat{}\,\square\,\hat{}\,\square문서, BBBR)$ ⑩

그림 6. PDA를 통한 색인 방법 예제

단계 ⑨에서 "바이러스"를 색인어로 추출하고 스택 심볼이 G이므로 상태를 q_3로 변환하여 최종 수락 단계로 변환하여 색인어 "컴퓨터"를 추출하는 것과 계속해서 pop을 수행하여 복합어 "컴퓨터 바이러스"를 색인어로 추출하는 과정으로 나누어진다.

이와 같이 *EOL*까지의 모든 문자를 읽고 pop의 과정을 완벽하게 수행한다면 위의 문장"컴퓨터 바이러스나 백신에 대한문서"에 대한 색인어로 "컴퓨터", "바이러스", "컴퓨터 바이러스", "백신", "컴퓨터 바이러스 백신", "문서", "컴퓨터 바이러스 백신 문서" 등이 색인어로 추출되게 된다. 이는 KTSET에서 문서 검색을 위해 요구하는 "컴퓨터", "바이러스" 및 "백신"등이 색인어로 문서의 내용을 대표하고 불용어를 치환하였기 때문에 불필요한 색인어는 없다. 어절수에 관계없이 모든 색인어 추출이 가능하게 된다.

4. 성능평가

제안하는 색인법의 성능을 인터넷 기반에서 평가하기 위해 Intel 계열의 PentiumII 300Mhz PC에 Microsoft사의 IIS 4.0을 웹 서버(web server)로 ASP(active server page)로 구현하였다. 다음의 그림 7은 넷스케이프 4.72에서 실험한 결과를 보여주고 있다. 입력을 위한 구현 부분은 생략 되었다.

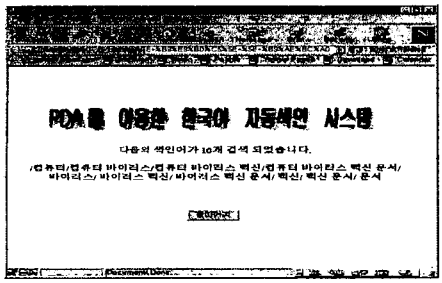

그림 7. PDA를 통한 색인어 추출

그림 7은 3.3절 예제 "컴퓨터 바이러스나 백신에 대한문서"이라는 텍스트에서 색인어로 10개가 추출되었음을 나타내고 있다. 앞 절에서 설명되었던 것처럼 "컴퓨터"라는 단일명사와 "컴퓨터 바이러스"라는 복합명사를 색인어로 추출되었다. 다음의 그림 8는 세 개의 명사로 이루어진 문장에서 색인어 추출 결과를 보여준다.

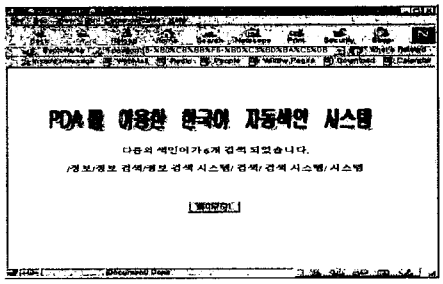

그림 8. 세 개의 합성명사 색인

그림 8는 "정보 검색 시스템"이라는 세 단어 합성명사에 대한 색인 결과를 보여주고 있다. 결과에서 알 수 있듯이 "정보", "검색", "시스템"등의 단일명사와 "정보 검색", "검색 시스템"등의 두 단어 합성명사와 함께 "정보 검색 시스템"등의 세 단어 합성명사의 색인이 가능하다.

다음 그림 9은 영어 문장에서 색인어를 추출한 결과를 보여주고 있다. 입력 문장으로 "korean automatic indexing system using the PDA"이 입력되었다고 가정한다.

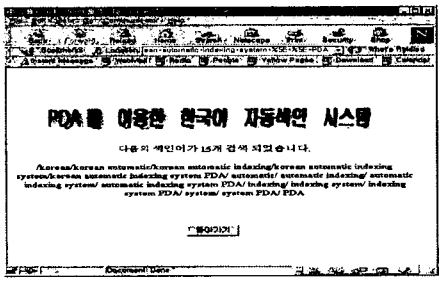

그림 9. 영어 문장에서 색인 결과

그림 9의 결과에서는 국문법을 바탕으로 색인어를 추출한 결과이므로 영문법을 적용하여 PDA를 수정한다면 영문의 색인어 추출이 가능할 것이다. 현재 인터넷과 정보통신에 대한 관심이 증가되고 있어, 인터넷을 통한 다른 언어로의 확장성은 검색 시스템에서 상당히 중요한 특성이 될 수 있다. 이상의 결과에서 알 수 있듯이 제안하는 시스템은 단일명사와 복합명사의 색인과 인터넷과 다른 언어로의 확장이 가능한 결과를 볼 수 있다.

5. 결론

인터넷의 발달로 방대한 양의 문서에서 정보를 효율적으로 관리하고 이용하기 위하여 많은 정보 검색 시스템들이 개발되어지고 있다. 문서의 내용을 나타내는 색인어를 추출하는 색인어 추출시스템은 정보 검색 시스템 성능에 큰 영향을 미치게 된다. 본 논문에서는 기존의 한국어 자동색인 시스템의 문제점등을 언급하고, 정형화된 한국어 자동색인 시스템을 제안하였다. 기존의 단어 중심 색인법은 복합명사의 색인이 어렵고, 형태소 중심색인 방법은 해당하는 언어의 사전 지식이 많이 필요하고, 분석에 필요한 사전(dictionary)이 필요하고 사전의 유지와 보수에 많은 노력이 필요하다. 또한 형태소 중심 색인방법은 구현이 복잡하다. 가장 최근의 n-gram 중심 색인방법은 n 값을 어떻게 정하는 지의 객관적인 평가가 어렵고 이로 인한 색인어 추출 시스템의 효율이 저하되고 부정확한 색인어 추출로 저장공간의 낭비를 가져온다. 제안하는 PDA를 이용한 자동색인 시스템은 언어에 대한 사전지식이 필요하지 않고, 복합명사 및 새로 생성된 신조어의 색인이 가능할 뿐 아니라 인터넷으로의 확장과 다른 언어로 구현이 쉬운 장점을 가지는 확장성이 뛰어난 특성을 가진다. 성능평가에서 KTSET를 구현된 PDA를 이용한 한국어 자동색인 시스템으로 색인어 추출 테스트를 하였다.

참 고 문 헌

[1] 김판구, 조유근. "상호 정보에 기반한 한국어 텍스트의 복합어 자동색인." 한국정보과학회 논문지 제21권 제7호 (94.7)

[2] 이준호, 안정수, 박현주, 김명호 "한글 문서의 효과적인 검색을 위한 n-gram 기반의 색인방법." 정보처리학회지 제13권 제1호 (96)

[3] Lee, J. H., Cho, Y. H., Park, H. R.. (1999) n-gram-based for Korean text retrieval. Paper presented at Information Processing and Management 35, 427-441

[4] Cavnar, W. B., 1994 N-Gram-Based TextFiltering for TREC-2. In Proceedings of the Second Text Retrieval Conf(TREC-2) NIST Special Publication 500-215, 171-179.

[5] 강승식, 장병탁. "음절 특성을 이용한 범용 한국어 형태소 분석기 및 맞춤법검사기." 정보과학회지(B) 제23권 제5호 (96,5)

[6] Yahushi Ogawa, Toru Matsuda.. (1999) Overlapping statistical segmentation for effective indexing of Japanese text. Information Processing and Management 35 463-480.

[7] Bo-Hyun Yun, Yong-Jae Kwak, Hae-Chang Rim. Alleviating syntactic term mismatches in Korean text retrieval Information Processing & Management ,V.35 N.4 1999, 07- 01.

[8] Jeffrey D. Ullman 저., 정인정 역, "오토마타와 계산이론" 홍릉과학출판사. 1993

[9] Hopcroft, Jeffrey D. Ullman Introduction to Automata Theory, Languages, and Computation Pressed by John E. Addison-Wesley Pub Co. (April 1979)

[10] Peter Linz. An Introduction To Formal Languages And Automata Jones and Bartlett Publishers (1997)