한글 하이퍼텍스트 자동변환시스템의 설계 및 구현

안병익, 김재군, 김영환
한국통신 소프트웨어연구소 인공지능연구실

The Design & Implementation of Korean Hypertext Automatic Translator

B.I.Ahn, Jay.Kim, Y.W.Kim
AI Section, Software Research Laboratories, Korea Telecom
Tel: (02)526-5913, Fax: (02)526-5909
E-mail: biahn@aistar.kotel.co.kr

요 약

하이퍼텍스트는 문서검색 전산화의 새로운 대안을 제시하고 있으나 저작에 많은 시간과 노력이 요구되는 단점이 있다.

본 연구에서는 기존의 한글문서를 하이퍼텍스트 문서로 자동 변환하는 변환시스템을 설계, 구현하였다. 문서는 사용자가 제공한 부제목형식의 정규표현식(regular expression)으로부터 논리적 구조가 분석되며 문서분할, 형태소분석, 대표카드결정 및 링크생성의 과정을 거쳐 하이퍼텍스트 문서로 변환된다.

시험운용 결과 본 시스템은 대량의 한글문서를 적은 노력으로 실용성있는 하이퍼텍스트 문서로 자동 변환할 수 있음을 입증하였다.

I. 서론

비 순차적인 방법으로 문서를 검색할 수 있게 하는 하이퍼텍스트는 컴퓨터에 의한 문서검색의 새로운 대안으로 각광받고 있다[1]. 하이퍼텍스트 검색의 가장 큰 걸림돌은 저작(authoring)에 드는 노력과 비용이 너무 크다는 점이다. 전문적인 하이퍼텍스트 저작도구를 사용하더라도 저작자는 적어도 문서의 내용을 완전히 파악하고 있어야 하며 링크의 연결에 많은 시간이 요구되기 때문이다. 따라서 적은 노력으로 기존의 문서를 하이퍼텍스트 문서로 자동으로 변환하는 변환 시스템의 개발이 연구되어져 왔다.

하이퍼텍스트 자동 변환을 위해서는 문서의 논리적 구조(logical structure)가 파악되어야 하는데 기존에는 논리적 구조가 명시된 특수한 문서를 사용하거나[2] 수작업으로 문서에 꼬리표를 붙이는 방법[3]이 사용되어져 왔다. 그러나 전자의 방법은 그와 같은 문서가

보편화되어 있지 않고 후자의 방법은 비효율적이라는 단점이 있다. 본 연구에서는 지면구조(layout structure)만으로 주어진 문서 내에 내재된 개념들을 자동으로 파악하는 방법을 제안하였으며 이 방법에 의거하여 시스팀을 구현하고 시험하였다.

2장에서는 하이퍼텍스트와 하이퍼텍스트 변환방법에 대해 설명하였고 3장에서는 시스팀의 구현내용 및 원리를 설명하였으며 4장에서는 시험결과를 분석하고 5장에서 결론을 맺었다.

II. 배경

1. 하이퍼텍스트

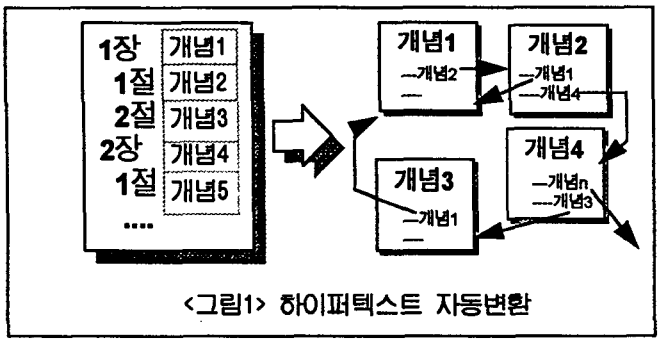
서적을 비롯한 많은 정보매체들이 선형적인 순서에 의해 정보를 저장하고 검색하는것과 달리 하이퍼텍스트는 인간의 연상작용과 유사한 방식인 비순차적인 순서로 자료를 서로 연결하여 정보를 얻는다. 하이퍼텍스트의 가장 중요한 요소는 카드와 링크로서 하이퍼텍스트는 이들의 망으로 이루어진다[4].

하이퍼텍스트 문서검색은 기존의 순차적인 문서검색에 비해 사용자가 찾는 내용을 쉽게 추적할 수 있고 텍스트 세그먼트를 여러 곳에서 참조할 수 있으므로 아이디어를 표현할 때 중복을 피할 수 있으며 화면에 동시에 여러 정보를 표현할 수 있다는 장점이 있다.

반면에 하이퍼텍스트는 비선형 구조의 문헌내에서 현재 위치와 추적방향에 대한 감각을 잃어버릴 수 있으며(Disorientation problem) 링크를 만들고, 이름을 부여하는데 필요한 인지적 오버헤드(Cognitive Overhead)에 익숙해지기 어려운 단점이 지적되고 있다. 여기에서 전자의 문제는 그래픽 브라우저나 정보검색 기법등을 이용하여 어느정도 해결할 수 있으나 후자의 문제는 하이퍼텍스트의 저작에 많은 노력과 비용을 필요로 하여 하이퍼텍스트 보급의 가장 큰 걸림돌이 되고 있다. 이 문제는 하이퍼텍스트 저작을 자동화 함으로써 해결할 수 있다.

2. 하이퍼텍스트 자동변환

기존의 텍스트문서를 하이퍼텍스트 검색을 위해 하이퍼텍스트 문서로 자동으로 변환하는 작업을 하이퍼텍스트 자동변환이라 한다.

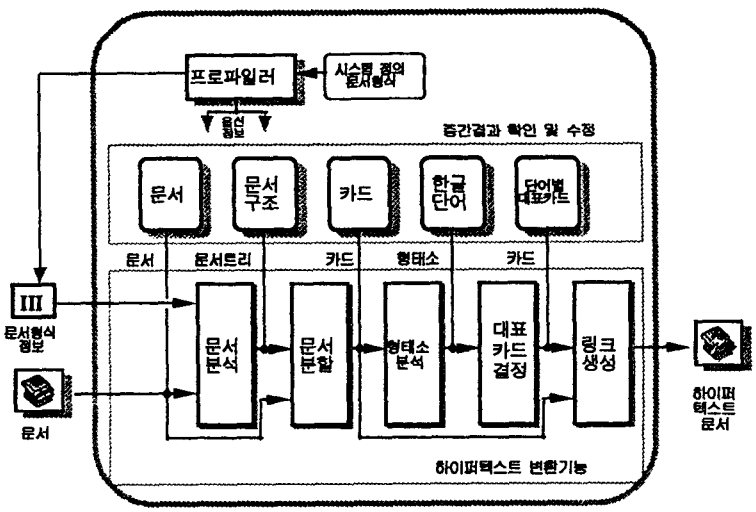


<그림1> 하이퍼텍스트 자동변환

<그림1>은 일반문서와 하이퍼텍스트문서의 차이를 도시한것이다. 하이퍼텍스트 문서는 문서내의 모든 개념이 각각 카드라는 단위로 분리되어 있으며 개념을 설명하는 내용 중에 다른 개념을 참조하는 단어가 나타나면 그 단어와 다른 개념간에 링크가 연결된다. 반면에 일반문서는 지면구조만을 갖고 있으며 문서가 표현하는 개념은 지면구조 속에 포함되어 있으나 명시되어 있지는 않다. 하이퍼텍스트 자동변환은 지면구조만으로 주어진 일반문서에서 개념을 추출하고 각각의 개념에 포함된 다른 개념의 참조단어를 발견하여 개념들 사이의 링크를 연결해 주는 과정이다.

문서상에서 개념의 추출은 문서의 지면 구조에서 논리적 구조(logical structure: 장, 절 등 단원의 계층적 구조)를 추출함으로써 실현될 수 있다. 이는 물론 논리적 구조의 구성원인 각각의 단원이 문서가 포함하고 있는 개념들을 대표한다는 가정이 전제된것이다. 문서의 논리적 구조 추출은 특수한 문서를 입력문서로 사용하는 방법과 수작업으로 문서에 꼬리표를 붙이는 방법이 사용되어져 왔다. 전자의 방법으로 사용될 수 있는 문서는 SGML (Standard Generalized Markup Language) 문서, 아웃라인 처리기(outline processor) 문서, 또는 TROFF, RTF(Rich Text Format)와 같은 편집문서등이며 이들의 문서에는 논리적 구조가 명시되어 있으므로 쉽고 정확하게 처리할 수 있다는 장점이 있으나 아직 이와 같은 류의 문서에 대한 표준화 및 보편화가 이루어지지 않아 적용이 비현실적이라는 문제가 있다. 후자의 방법은 사용자가 일일이 문서에 포함된 부제목 위치를 찾아서 특수문자를 이용하여 꼬리표를 붙인 후 프로그램이 이 꼬리표를 인식하여 논리적 구조를 추출하는 방법이다. 이 방법은 수작업이 수반되므로 비효율적이고 오류의 가능성이 있는 단점이 있다.

링크를 연결하는 방법으로서 정보검색의 방법을 적용할 수 있다. 즉, 각 카드를 정보검색의 문서로 보고 각각에 대해 형태소분석 및 자동색인을 통해 카드의 대표단어를 추출한 후 모든 카드에서 나타나는 단어들을 정보검색의 질의어와 같이 처리하여 정보검색 결과 한계값 이상의 가장 높은 문서값을 갖는 카드와 링크를 연결하는 방법이다. 여기에서는 시소러스의 사용등 정보검색의 다양한 기법들이 똑같이 적용될 수 있다[5].



<그림2> 시스템 블록다이어그램

III. 한글 하이퍼텍스트 자동변환시스템

1. 시스템 구성

본 시스템에서는 문서의 논리적 구조를 추출하기 위해 사용자가 각 수준에 해당하는 부제목의 형식을 정규표현식(regular expression)으로 정의하고 시스템이 문서에 대해 정규표현

검색을 수행하여 문서에 포함된 각 수준별 부제목들을 발견한 후 이를 토대로 문서트리를 구축함으로써 처리된다. 링크의 생성방법은 형태소분석 및 대표카드결정 방법을 사용한다.

본 시스템은 한글 MS-윈도우즈3.1 상에서 C++언어를 사용하여 구현하였다. <그림2>는 본 시스템의 블록다이어그램을 보인것이다. 시스템은 프로파일러(profiler)라 부르는 입력을 위한 부시스템과 분서분석, 문서분할, 형태소분석, 대표카드결정 및 링크생성등 5개의 기능 부시스템, 그리고 처리 단계별로 그 결과를 확인 수정할 수 있는 5개의 확인 및 수정 부시스템으로 구성된다. 다음에 이들 각각의 기능에 대해 설명하였다.

2. 프로파일러

프로파일러에서는 대화상자(Dialog Box)를 통하여 처리할 문서를 선택하고 그 문서의 부제목의 형식을 정규표현식으로 정의하며 여러가지 변환옵션을 입력한다.

부제목의 정규표현식은 부제목의 수준에 따라 5개 까지 지정할 수 있다. 예로서 한국통신의 연구 과제 제안서는 각 부제목의 형식으로서 부제목의 가장 윗 수준부터 I., 1., 가., 1), 가) 와 같은 형식을 취하도록 규정되어 있는데[6] 이들 각각의 정규표현식은 <표1>과 같이 정의할 수 있다.

부제목 수준	부제목	정규표현식
1	I.	^[\t]*[IVX]+\. .
2	1.	^[\t]*[0-9]+\. .
3	가.	^[\t]*(가|나|다|라|마|바|사|아|자|차|카|타|파|하)\. .
4	1)	^[\t]*[0-9]+\) .
5	가)	^[\t]*(가|나|다|라|마|바|사|아|자|차|카|타|파|하)\) .

<표1> 부제목의 정규표현식 예

이와 같은 정규표현식은 문서의 형식을 비교적 정확하게 정의할 수 있게하는 장점이 있으나 일반 사용자에게는 입력이 복잡한 단점이 있다. 이 문제점을 덜기위한 방법으로서 본 시스템에서는 한글문서에서 나타날 수 있는 대부분의 제목형식에 대한 정규표현식을 미리 입력하여 사용자가 선택할 수 있다.

I.	제 1 장	제 1 절	1.	가.	1.1	1.1.1	1.1.1.1	i.	1)	가)

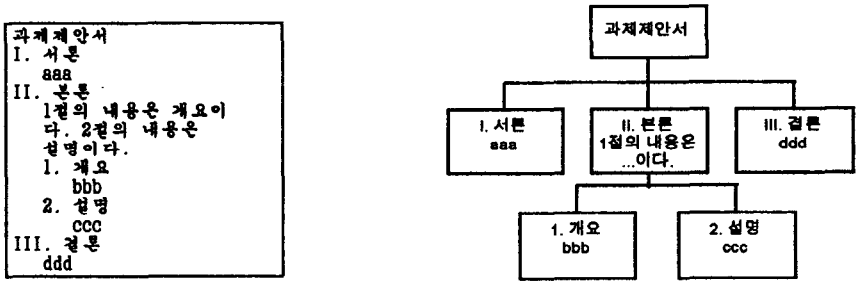
<표2> 시스템 정의 부제목의 유형

2. 문서분석

프로파일러에서 선택한 문서와 부제목 형식정보를 토대로 문서의 지면 구조에서 문서의 논리적 구조를 추출한다. 먼저 정규표현 검색을 통해 문서에서 각 수준의 부제목의 내용 및 위치를 인식하고 이 결과를 토대로 문서의 논리적 구조를 표현하는 문서트리를 구축한다. 예로서 <그림3(가)>와 같은 간략한 문서는 <그림3(나)>와 같은 문서트리로 구성될 수 있다.

시스템 구현 후 여러 종류의 문서를 변환하는 과정에서 정규표현 검색으로 명확하게 처리할 수 없는 예를 발견할 수 있었다. <그림3(가)>의 내용 중 본론의 내용에서 첫번째 문장을 고려해보자. 이 문장의 마지막 부분인 '다.'는 문장이 시작된 줄을 넘어서 다음줄의 첫번째 부분에 나타나고 이어서 두번째 문장이 시작된다. 따라서 두번째 줄은 '다. 2절의 내용은'이 되며 이 줄은 <표3>의 세번째 부재목의 정규표현식에 부합하여 부제목으로 잘못 인식하게 된다. 이 문제는 정규표현식의 수정으로도 처리할 수 없으므로 다음과 같은 후처리 알고리즘을 적용하여 해결하여야 한다.

(1) 부제목이 '다.'로 시작되면 같은 수준의 전후 부제목을 검사하여 전의 부제목이 '나.'로 시작되지 않거나 후의 부제목이 존재하고 '라.'로 시작되지 않으면 이 줄은 부제목이 아니다.
(2) 바로 앞 줄의 마지막과 병합하였을 때 '다.'로 끝나는 어미를 형성하지 않으면 이 줄은 부제목이다.
(3) (1)과 (2)의 두 경우를 모두 만족하면 이 줄은 부제목이 아니다.



(가) (나)

<그림3> 문서트리

3. 문서분할

문서트리의 각 노드는 하나의 개념을 표현하므로 하나의 카드로 분할 할 수 있다. 문서분할기능 부시스템에서는 문서트리를 토대로 문서를 카드로 분할한다. 분할된 각각의 카드의 내용은 제목, 본문, 부제목의 3구역으로 구분된다. 그 예로서 <그림3(나)>에서 중앙의 'II. 본론' 노드에 해당하는 카드는 다음과 같은 내용을 갖는다.

제목	II. 본론
본문	1절의 내용은 개요이다. 2절의 내용은 설명이다.
부제목	1. 개요
	2. 설명

<표3> 카드의 내용 예

하이퍼텍스트 카드는 컴퓨터의 화면상에서 검색되므로 그 크기는 한 화면의 분량을 지나치게 초과하는 것은 바람직하지 못하다. 따라서 위의 과정으로 분할된 카드중 그 크기가 큰

카드에 대해서는 추가로 분할한다. 카드의 크기는 카드내용의 3구역을 합한 줄의 수로 규정하며 분할은 본문 내에서 이루어진다. 본문의 분할은 본문 내에서 문단(paragraph)을 인식하여 문단 단위로 분할하는데 문단은 다음과 같은 과정을 통해 인식한다.

(1) 본문중 공백줄이 있으면 위와 아래는 다른 문단이다.
(2) 3개 이상의 연속적인 줄이 같은 열에서 첫 글자가 시작되면 그 열을 좌측마진으로 규정한 후 좌측마진 이후의 열에서 시작되는 줄은 새로운 문단의 첫줄이다.

한 화면의 크기를 p로 규정하였을 때 분할될 부분은 p번째 줄 이후에 처음 나타나는 새로운 문단이며 분할하고 남은 부분은 같은 알고리즘을 회귀적(recursive)으로 반복 적용한다.

추가로 분할된 카드는 모두 같은 제목과 부제목을 갖는다.

4. 형태소분석

문서가 카드로 분할된 이후 링크를 연결하기 위한 첫번째 과정으로 각 카드를 대표할 수 있는 단어를 추출하는데 이는 한국어 정보검색방법인 형태소분석과 자동색인을 통해 처리할 수 있다. 카드의 내용을 형태소 분석하여 카드에 포함된 모든 단어를 추출하고 단어의 빈도수에 의한 자동색인을 통해 중요도가 가장 큰 몇개의 단어를 대표단어로 결정하는 방법이다. 그러나 현실적으로 소규모의 텍스트에 대한 적절한 자동색인 방법이 없다는 문제가 있다. 한국어 정보검색에서 자동색인의 대상은 문서 전체의 형태소에 대하여 행하여지기 때문에 단어빈도수에 의한 자동색인이 가능하지만 카드의 형태소는 그 수가 적기 때문에 단어별 빈도수의 편차가 적어서 빈도수에 의한 자동색인 효과가 거의 없어서 대표단어 선정에 어려움있다.

본 시스팀에서는 이에 대한 대안으로 카드의 제목만을 형태소분석하고 자동색인 과정은 생략하였다. 이 방법은 대표단어의 수를 격감시켜 링크의 수가 많지 않게 되는 문제가 있으나 대부분의 경우 카드의 제목이 카드의 내용을 대표하기 때문에 대표단어를 효과적으로 추출할 수 있는 장점이 있다. 형태소분석기는 한국과학기술원에서 한국통신의 지원으로 개발된 프로그램을 본 시스팀에 맞게 수정하여 사용하였다[7].

5. 대표카드의 선정

한 문서에서 분할된 여러개의 카드 각각에 대해 대표단어를 추출하면 하나의 단어가 복수의 카드의 대표단어로 사용된 경우가 발생할 수 있다. 대표단어는 하이퍼텍스트 링크의 착점으로 사용되고 착점은 유일하기 때문에 이 경우 복수의 카드중 하나를 선정하여야 한다. 대표카드는 복수의 카드 각각에 대한 가중치를 구한 후 가중치가 가장 큰 카드로 결정한다. 본 시스팀에서는 카드의 가중치를 계산하기 위해 다음 식과 같은 휴리스틱을 적용하였다.

$$\text{단어 } W\text{에 대한 카드 } C\text{의 가중치} = C\text{에서 } W\text{의 빈도수} * \alpha + C\text{의 제목수준} * \beta$$

α, β는 상수이며 사용자가 옵션에서 지정할 수 있고 초기에 각각 2와 1의 값을 갖는다. 이 식에서 β의 초기값으로 카드의 제목수준이 낮을수록 큰 가중치를 갖게 한 이유는 제목수준이 낮을수록 카드 내용에 단어에 대한 보다 구체적인 설명이 포함되었을 것이라는 가정이 적용된것이다.

6. 링크의 연결

각 카드의 내용에서 모든 카드의 대표단어를 검색하여 발견되면 링크를 연결하여 하이퍼텍스트 문서를 생성한다. 즉, 카드의 내용중 카드 C의 대표단어 W가 발견되면 문자열 W에서 카드 C로 링크를 연결한다. 이 하이퍼텍스트 문서는 Microsoft사에서 규정한 RTF 화일형식으로 기록된다. 이 화일은 동사에서 제공하는 도움말컴파일러 프로그램을 사용하여 윈도우즈 도움말 화일형식의 하이퍼텍스트 문서로 변환할 수 있으며[8] 윈도우즈에 기본으로 제공되는 도움말 프로그램으로 검색할 수 있다.

7. 확인 및 수정 부시스팀

하이퍼텍스트 자동변환 과정은 각각이 오류의 가능성을 포함하고 있거나 처리성능에 따라 그 결과가 달라질 수 있기 때문에 본 시스팀에서는 각 처리과정에 의해 생성된 중간결과를 확인하고 수정할 수 있도록 하였다. 이를 통하여 하이퍼텍스트 변환의 전 과정을 수작업으로 처리할수도 있기 때문에 하이퍼텍스트 저작도구로 전용하여 사용할 수도 있다.

IV. 구현결과 분석

서로 다른 유형의 20개의 문서를 선별하여 시험한 결과 부제목의 형식은 96%가 시스팀에서 정규표현식을 제공하는 11개의 부제목에 포함된 유형이었으며 따라서 자동화를 위한 입력작업에 큰 어려움이 없었다. 문서분석기능은 93%의 정확도를 보여 정규표현식을 사용한 문서분석방법이 효과적임을 보여주었다. 문서의 추가분할은 75%의 정확도로서 카드의 내용에서 문단의 인식 방법에 개선의 여지가 남아 있음을 보여 주었다. 대표단어의 추출은 카드의 제목만으로 형태소분석을 한 결과 수작업으로 선정한 단어와 약 55%의 일치율을 보였으며 4%의 카드는 대표단어가 없는것으로 인식하였다. 이는 '개 요'와 같이 단어 사이에 공백을 넣은 부제목이 포함되어 있는 경우에 형태소 분석을 제대로 처리하지 못한 결과로 풀이된다. 전체 대표단어중 36%가 복수개의 카드에서 대표단어로 추출되었다. 한 카드내에서 평균 링크 수는 4.6개였으며 문서의 내용일 일반적인 내용일때 보다 전문적인 내용인 경우 링크의 수가 더 많았다.

자동변환에 걸리는 시간은 평균 14페이지의 문서를 문서분석에서 링크생성까지 약 100초 정도의 시간에 처리하였으나 도움말 컴파일에 평균 6.3분의 시간이 걸려서 문서변환의 평균시간은 약 8분이 소요되었다. 도움말 컴파일러 수행에 많은 시간이 걸린것은 이 프로그램 고유의 문제인것으로 분석되며 별도로 이 프로그램을 구현하거나 하이퍼텍스트 검색기를 독자적으로 구현함으로써 처리속도 향상시킬수 있을 것으로 기대된다.

V. 결론

본 연구에서는 하이퍼텍텍스트를 자동으로 변환하는 방법으로서 정규표현식을 이용하여 문서의 논리적 구조를 검색하는 방법을 제안하였으며, 자동한글 하이퍼텍스트 자동변환시스팀을 구현하고 시험하였다. 정규표현식은 문서의 지면구조에서 논리적 구조를 추출하는 효과적인 수단이며 이를 정의 하는데 따르는 어려움은 한글문서에서 빈번히 출현하는

부제목의 형식을 시스템에서 제공함으로써 해결할 수 있다. 대표단어 선정에 빈도수를 사용한 자동색인방법 적용이 어려우므로 짧은 텍스트를 색인할 수 있는 새로운 자동색인기술 개발이 요구된다. 중복되는 대표카드는 단어의 빈도수와 부제목의 수준을 인수로 하는 휴리스틱 공식을 적용하여 제거할 수 있다. 보다 형태소분석기의 성능개선과 소규모 텍스트의 색인방법은 추후에 연구하여야 할 과제이다.

시험운용 결과 본 시스템은 대량의 한글문서를 적은 노력으로 실용성있는 하이퍼텍스트로 자동 변환할 수 있으며 현재 한국통신 소프트웨어연구소의 연구과제 보고서 검색에 활용하고 있다.

참고문헌

[1] Jakob Nielsen, 'Hypertext & Hypermedia,' ACADEMIC PRESS, 1990.

[2] Roy Rada, 'Hypertext : From Text To Expertext,' McGraw-Hill, pp9~12, 1991.

[3] 김명관, '한글 하이퍼텍스트 자동생성기,' 한국 전자통신연구소, 1992

[4] 이성호, '하이퍼텍스트: 개요, 구조 및 응용분야,' 전기통신연구, 한국통신, pp87~96, 1992. 12.

[5] Mark Frisse, 'From Text to Hypertext,' BYTE Magazine, pp247-253, Oct. 1988.

[6] 한국통신 기술기획실, ''93연구과제 제안지침,' 한국통신, pp37~46, 1992. 6.

[7] 최기선 외, '지능형 정보검색에 관한 연구,' 2차년도 최종연구보고서, 한국통신, 1992. 12.

[8] Microsoft, 'Microsoft Windows Software Development Kit,' Microsoft Corp., 1990.