동적 정보 저장을 위한 자동 하이퍼텍스트 색인 기법의 개발

이 동 애[†] · 장 덕 성[††]

요 약

하이퍼텍스트 정보를 저장할 때 정보가 삽입, 삭제, 변경되면, 인접한 정보들에 대한 하이퍼텍스트 링크도 변화되어야 한다. 하이퍼텍스트 링크는 하이퍼텍스트 색인어를 기준으로 관련있는 다른 정보를 찾는 수단을 제공한다. 따라서 하이퍼텍스트 색인어를 관리하는 것이 동적 정보 저장의 핵심이 된다. 본 논문에서는 새로운 정보가 삽입, 삭제, 변경될지라도 시스템이 안정성을 유지하며, 변경된 부분에 대한 하이퍼텍스트 색인어와 하이퍼텍스트 링크가 동적으로 결정될 수 있는 방법을 제시한다. 이를 위해 동적 색인기를 만들고, 동적 색인기의 동작을 돕기 위해 색인어 사전, 불용어 사전, 조사 사전, 역색인 파일, 시소러스 등을 구성한다.

Development of an Automatic Hypertext Indexer for Dynamic Information Storage

Dong-Ae Yi[†] · Duk-Sung Jang[††]

ABSTRACT

The hyperlinks to related nodes should be changed when we insert, delete, or modify an information in a hypertext database. We can find more informations by means of hyperlinks that are based upon hypertext indexes. Therefore, the management of the hypertext indexes is an important component for dynamic information storage. In this paper, we suggest a method to manage the hypertext indexes and to determine hyperlinks automatically by using a dynamic indexer. We also construct index, stopword, and postposition dictionaries, an inverted index file, and a thesaurus to help the dynamic indexer.

1. 서 론

정보 검색의 대상이 되고 있는 정보들 즉 텍스트, 이미지, 소리, 그래픽, 소프트웨어 루틴 등은 대부분 수시로 삽입, 삭제, 변경되어야 하는 동적 정보들이다. 이러한 동적 정보들을 하이퍼텍스트 시스템으로 구성하게 되면, 어떤 정보의 변경이 연관되는 다른 정보들에도 영향을 미치게 되므로 삽입, 삭제, 변경을 대비한 자료구조와 검색방법이 고려되어야 한다. 하이퍼텍스트 시스템은 정보의 상호연관성을 중심으로 구조화되어 있고 이들을 비선형적으로 접근하도록 되어 있기 때문에, 하이퍼텍스트 링크를 동적으로 관리되는 것이 결코 쉬운 일이 아니다. 종래의 정보검색은 키워드를 얼마나 잘 관리하느냐에 따라 탐색의 성공여부가 결정되었지만, 하이퍼텍스트 시스템에서는 키워드의 변화를 내부적으로 관리해야 할 뿐아니라 시각적으로 보여주어야 하며, 상호 연결된 이웃 자료

※이 논문은 1995년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

† 정 회 원:계명대학교 전자계산학과
†† 정 회 원:계명대학교 전자계산학과 교수

논문접수:1996년 11월 13일, 심사완료:1997년 8월 29일

들에 대한 링크를 수정하여야 하기 때문이다[11].

본 논문에서는 하이퍼텍스트 정보검색에서, 새로운 정보가 삽입되거나 기존의 정보가 삭제 또는 변경되더라도 시스템이 안정성을 유지하면서, 변경된 정보에 대한 하이퍼텍스트 색인어와 하이퍼텍스트 링크가 동적으로 결정될 수 있도록 하는 방법을 연구한다.

정보 검색에 관한 연구들을 살펴보면 주로 대규모 정보에 대한 검색[16, 21], 하이퍼텍스트 응용에 관한 연구[3, 4, 17, 18], 혹은 검색 효율을 높이기 위한 한국어 특성에 관한 연구[1, 2, 6, 7, 8, 12, 13, 14] 등으로 나누어 볼 수 있는데, 최근에는 인터넷을 통한 정보검색[20] 혹은 다자간 공동 작업이 핫 잇슈로 떠오르고 있다. 다자간 공동작업 혹은 인터넷에서는 정보의 변경이 더욱 빈번히 일어나므로 동적 정보 저장시에 색인어를 관리하는 문제가 대단히 중요한 과제로 대두되는 바, 본 연구에서는 다자간 공동 작업시 각 팀에서 개발한 소프트웨어 루틴의 공유를 목적으로, 소프트웨어 루틴들을 대상으로 하는 하이퍼텍스트 검색에 관한 연구를 수행하였다.

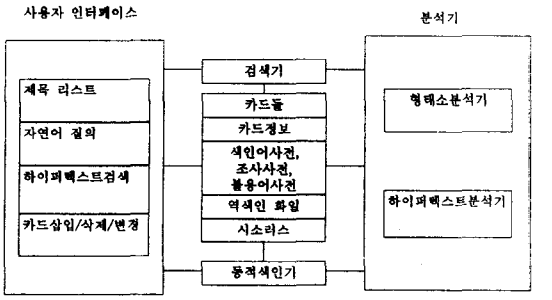
소프트웨어 루틴 검색은 소프트웨어 생산성을 높이고자 관련 소프트웨어 루틴들을 특별한 방법으로 구성하여 두고 필요한 루틴을 검색할 수 있도록 한 것이다. 이에 관한 연구가 여러 연구기관에 의해 수행되었다[5, 9]. 본 논문에서 제안하는 방안이 기존 연구와 다른 점은 각 소프트웨어 루틴에 설명문을 두고, 그 설명문을 근거로 색인어 리스트와 하이퍼텍스트를 구성하여 자연어 질의 및 하이퍼텍스트 질의가 가능하도록 하고, 이들 각각의 루틴들이 동적으로 추가·삭제·변경되더라도 동적 색인기가 자동으로 색인어의 변경을 관리하도록 한 것이다.

제 2장에서는 전체 시스템의 개괄적 구조를 다루고, 제 3장에서는 동적 색인기, 제 4장에서는 검색기에 대해 설명하고, 제 5장에서 결론을 맺는다.

2. 시스템의 구조

각 소프트웨어 루틴은 소스 코드와 함께 제목, 설명문이 함께 존재한다. 루틴의 추가·삭제·변경은 소스 코드, 제목, 설명문을 묶어 하나의 정보 단위로 처리되므로 이것을 카드(card)라고 명명한다. 본 시스템은 두가지 모드(mode)가 필요하다. 하나는 카드의 추가·삭제·변경을 위한 모드이며, 다른 하나는 검색 모드이다. 추가·삭제·변경 모드는 카드의 추가·삭제·변경 요구가 있을 때, 해당 카드의 설명문에서 색인어들을 추출하여 역색인 파일을 변경하고 설명문의 색인어들을 마크업한다. 검색 모드는 자연어 질의 혹은 하이퍼텍스트 질의를 통해 카드들을 조회하여 브라우징할 수 있게 한다.

추가·삭제·변경을 위해서는 설명문을 형태소 분석하여 색인어를 추출해야 하는데, 이 과정에서 조사사전, 불용어사전, 색인어사전, 시소러스를 참조하게 된다. 추출된 색인어들은 동적색인기를 거쳐 역색인 파일을 변경한다. 하이퍼텍스트 분석기는 변경된 설명문에서 바뀐 색인어들을 찾아 새로 하이퍼링크될 색인어들을 등록하거나 말소시킨다.



(그림 1) 시스템의 전체 구성도
(Fig. 1) System overview

검색모드에서는 자연어 질의문과 각 카드의 설명문간의 유사도를 계산하여 임계값 이상의 카드를 중요도 순으로 보여 주고 사용자로 하여금 하나를 선택하게 한다. 하나의 카드가 발견되면 그 다음부터는 하이퍼텍스트 검색이 가능하다. 그러나 일반적인 하이퍼텍스트 검색과 다른 점은 무조건 하이퍼 링크를 따라가는 것이 아니라, 하이퍼색인어와 가장 유사도가 높은 카드를 찾기 위해, 하이퍼 색인어와 각 카드간의 유사도를 계산하여 유사도가 높은 순으로 후보 카드들을 찾아내게 된다. 실제로 각 카드를 찾아가기 위해서는 각 카드들의 카드의 시작 위치, 다른 카드에 대한 하이퍼링크 등의 정보를 담은 카드 정보도 필요하다. 시스템의 전체 구성도는 (그림 1)과 같다.

3. 동적 색인기

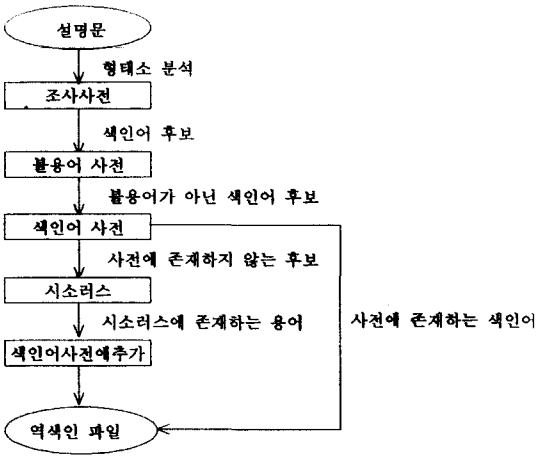
3.1 각종 사전과 시소러스

조사 사전은 한 글자로 된 조사 '은', '는', '이', '가' ... 에서부터 아홉글자로 된 조사 '에게서부터조차라도', '한테부터만이야말로' ... 까지 길이순으로 185개를 저장한 것으로서 색인어 후보를 추출하기 위해 형태소 분석시에 사용한다.

불용어 사전은 색인어로 가치가 없는 명사 혹은 동사를 걸러내기 위해 '프로그램', '알고리즘', '결과' ... 등과 '이용하다', '구현하다', '보여주다' ... 등, 평범하게 사용되는 고빈도 용어들을 포함한다. 변경 사항이 있는 카드의 설명문에 대해 형태소 분석이 끝나고 색인어 후보가 추출되면 제일 먼저 불용어 사전을 검사한다. 불용어 사전에 있는 용어는 색인어 대상에서 일단 제외되고, 불용어 사전에 존재하지 않는 용어들에 대해 색인어 사전을 참조한다.

색인어 사전은 색인어 후보들 즉, 명사, 명사-명사, 형용사-명사, 형용사-명사-명사, 명사-명사-명사 형태를 취하는 것 중 단일 명사, 혹은 복합 명사를 추출할 때 사용된다. 예를 들어 '정돈된 자료 형태'라고 했을 때 색인어 사전에 '자료'만이 등록되어 있으므로 '자료'를 색인어로 삼는다. 또 '자료 구조'라고 하면 색인어 사전을 참조하여 '자료구조'를 복합 명사로 인식한다. 다시 말하자면 빈칸으로 구분된 각각의 용어를 색인어 사전에서 찾아보고 두 개 이상의 용어가 발견되면 두(혹은 세) 용어를 합쳐 복합 명사로 간주하여 다시 사전을 찾는다. 만약에 그러한 복합 명사가 사전에 존재하지 않으면 두(혹은 세) 용어 각각을 모두 색인어로 취한다. 색인어 사전에는 색인어로 사용될 용어들이 수시로 등록되고 삭제되고 변경될 수 있도록 구성되고, 또한 빠른 검색이 보장되어야 하므로 해쉬 테이블로 구성하였다. 어떤 용어가 만약 색인어 사전에 존재하면 그 용어를 색인어로 인정하여 색인어에 대한 역색인 파일을 구성하는데 사용된다. 만약 색인어 사전에 존재하지 않으면 동의어가 아닌지 검사하기 위하여 시소러스(동의어 사전)를 참조하여, 시소러스에 있는 대표어와 함께 색인어 사전에 추가시킨다. (그림 2)는 설명문에 대해 자동 색인하는 과정을 보인 것이다.

문장 분석에 있어서 구문 해석은 구현하기 매우 복잡하여 연구의 대상은 되고 있지만 실제 정보 시스템에 이용하기에는 아직 이르고, 형태소 분석은 이미 그 기술이 보편화되어 있고 구현이 용이하다. 본 논문에서는 색인어들을 완벽하게 찾아내는 것보다는 기존 색인어에 변화가 있을 때 그것을 관리하는 동적 색인이 목적이기 때문에, 단순하게 형태소 분석만으로 색인어 후보들을 추출한다. 앞서 언급한 것처럼 형태소 분석이후, 불용어 사전에 의해 색인어 후보가 찾아지면 색인어 사전에 있는 용어들로 색인어를 구성한다. 색인어 사전에 등록되지 않은 용어들은 시소러스를 통해 새로운 색인어로 인정된다.

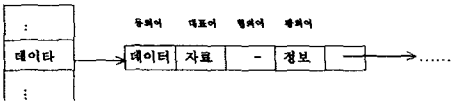


(그림 2) 자동 색인 과정
(Fig. 2) Dictionary reference flow for automatic indexing

본 연구에서는 컴퓨터 대사전[15]을 참조하여 소프트웨어 전반에 걸친 대부분의 용어를 정리하여 시소러스에 저장하였다. 그리하여 색인어 사전에 존재하지 않는 후보 색인어가 대표어와 유사어(동의어, 광의어, 협의어 등을 모두 포함하여 일컫는 용어)로 판정되면, 대표어와 꼭같은 것으로 취급하여 색인어 사전에 등록한다. 많은 논문에서 시소러스에 대한 구성을 이진 트리를 사용하고 있지만[6, 12], 본 논문에서는 (그림 3)의 b)와 같이 해쉬 테이블로 구성하였다. 입력은 (그림 3)의 a)와 같은 형태로 입력하면 시소러스 관리기가 각 동의어를 '가', '나', '다' ... 군으로 나누어 'ga_thesaurus', 'na_thesaurus', 'da_thesaurus' ... 라고 이름한 시소러스 테이블에 저장한다. 각 시소러스 테이블은 99개의 버켓(bucket)을 갖는다.

대표어	동의어수	BT수	NT수	동의어들	BT들	NT들
자료	2	1	0	데이터, 데이타	정보	-
자료형	2	1	10	데이터형, 데이타형	형	정수, 실수, 문자, 스트링

(a) 시소러스 입력 양식

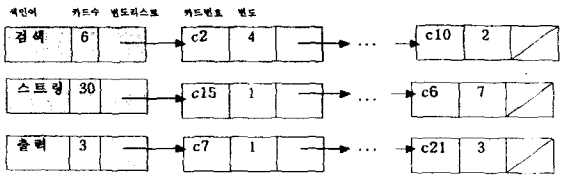


(b) 'da_thesaurus'의 구조

(그림 3) 시소러스의 구성
(Fig. 3) Construction of a thesaurus

3.2 역색인 파일

역색인 파일에는 색인어, 카드수, 그리고 빈도 리스트에 대한 포인터가 저장된다. 카드수는 해당 색인어를 갖는 카드가 몇개인지 저장한다. 빈도 리스트를 구성하는 각 노드들은 해당 색인어를 갖는 각 카드번호와 해당 색인어의 개수, 다음 노드의 주소로 구성된다. 색인어들은 자주 추가·삭제·변경이 일어나므로 리스트로 구성하였다. (그림 4)는 역색인 파일의 구조를 보여주고 있다.



(그림 4) 역색인 파일의 구조
(Fig. 4) Structure of an inverted index file

3.3 동적 색인기

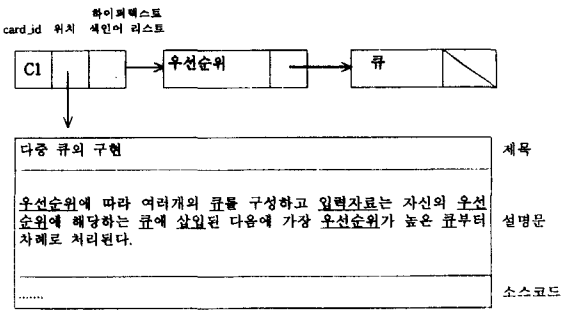
카드의 설명문에서 색인어를 추출해서 역색인 파일을 구성한다. 카드들이 추가·삭제·변경될 때마다 추가되어야 할 새로운 색인어, 삭제되어야 할 색인어, 변경되어야 할 색인어가 생긴다. 형태소 분석기가 카드의 색인어를 찾아내고, 동적 색인기가 새로운 색인어를 등록하고, 삭제해야할 색인어를 제거하여, 역색인 파일의 내용을 변화시킨다.

- 추가시 : 새 카드가 삽입되면 설명문을 형태소 분석하여 색인어를 찾아낸다. 동적 색인기는 이 색인어를 갖고 역색인 파일을 참조한다. 만약 역색인 파일에 해당 색인어가 존재하면 카드의 수를 1 증가시키고, card_id#와 빈도수를 빈도 리스트(frequency list)에 저장한다. 여기서 빈도수란 각 카드가 가지고 있는 해당 색인어 갯수를 의미한다. 만약 역색인 파일에 색인어가 존재하지 않으면 새로운 색인어 노드를 만들고, 또한 새 card_id#와 빈도수 노드를 생성해서 연결한다. 하이퍼텍스트 분석기는 설명문의 각 색인어를 역색인 파일의 각 색인어와 대응시켜 보고 같은 것이 있으면 자동으로 마크하여 하이라이트하고, 새 카드의 색인어들을 배열 속에 가나다 순으로 담아 동적 색인기에 넘겨준다. 이 배열을 '하이퍼텍스트 색인어 집합'이라고 한다. 동적 색인기는 새 카드에 대한 정보를 카드 정보에 추가할 때 새 카드에 대한 포인터뿐 아니라 하이퍼텍스트 색인어 집합에 대한 포인터도 함께 마련한다.
- 삭제시 : 한 카드가 삭제되어 없어지면, 동적 색인기는 먼저 하이퍼텍스트 색인어 집합을 취하여 역색인 파일의 해당 색인어를 찾아 카드 갯수를 1 감소하고 해당 카드의 빈도 노드를 삭제한다. 만약 카드 갯수가 0이 되면 그 색인어를 역색인 파일에서부터 제거한다. 다음으로 카드 정보의 해당 카드를 찾아 제거하고, 실제 카드도 데이터베이스에서 삭제한다. 실제 카드의 삭제는 카드 정보에 card_id#와 카드 위치가 있으므로 해당카드를 찾아 제거할 수 있다.
- 변경시 : 카드 내용이 변경될 때는 하이퍼텍스트 색인어 집합에서 삭제되어야 할 색인어와 추가되어야 할 색인어를 발견하여 변경시켜야 한다. 하이퍼텍스트 분석기가 변경된 색인어 집합을 참조하여 해당 카드를 다시 마크업한다. 또한 역색인 파일에서 변경된 색인어를 찾아 빈도 리스트를 수정하여야 한다. 만약 색인어가 없어졌다면 그 색인어의 빈도수는 감소될 것이며, 색인어가 추가적으

로 더 많이 생겼다면 빈도수는 증가할 것이다. 색인어의 빈도수가 감소되어서 0이 되면 빈도 리스트에서 그 노드는 삭제 된다. 물론 새로운 색인어가 생길 수도 있다. 이때는 역색인 파일에 새로운 색인어로 등록되고 카드 수는 1이 되며, 빈도 리스트는 오직 하나의 노드만을 가지게 된다.

3.4 카드 관리

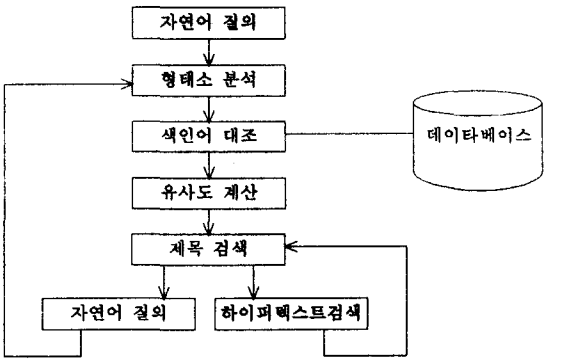
카드 정보는 실제 카드가 위치한 장소를 나타내는 위치와, 하이퍼텍스트 색인어 집합을 갖는다. 형태소 분석에 의해 발견되는 모든 명사(및 동사)들 중 시소러스에 존재하는 단어들이면 모두 색인어 대상이 되어 색인어 사전에 등록된다. 그러나 그것들이 모두 하이퍼텍스트 색인어가 되는 것은 아니다. 문장 분석기가 새 단어를 발견하게 되면 색인어 사전에 등록하고 그것을 하이퍼텍스트 분석기에 넘겨주어, 사용자에게 하이퍼텍스트 색인어로 등록할 것인지 묻게 된다. 사용자가 하이퍼텍스트 색인어로 선택한 몇 개의 단어만을 하이퍼텍스트 색인어 리스트에 넣고 그 단어들을 하이라이트 한다. (그림 5)는 카드 #1이 갖는 설명문과 하이퍼텍스트 색인어 리스트를 보여주는 그림이다.



(그림 5) 카드 정보의 예
(Fig. 5) Example of a card information

4. 검색기

검색은 자연어 검색과 하이퍼텍스트 검색이 가능하다. 자연어 질의에서는 질의문에 대한 형태소 분석



(그림 6) 검색과정
(Fig. 6) Retrieval flow

으로 색인어를 찾고 설명문의 색인어와 대조 작업으로 유사도를 계산하여 원하는 카드를 찾게하는 것이다. 하이퍼텍스트 검색은 하이퍼링크될 색인어와 각 카드의 설명문간의 유사도를 계산하여 유사도가 높은 순서대로 사용자에게 보여주고, 사용자가 검색된 후보카드들의 제목중에서 하나를 선택하여 카드의 내용을 브라우징하면서 탐색할 수 있게 한 것이다. 자연어 검색과 하이퍼텍스트 검색은 언제든지 사용자가 선택하여 사용할 수 있다.

기존의 하이퍼텍스트 시스템은 사용자가 직접 개입해서 정적으로 하이퍼텍스트 색인어를 마크업하고 하이퍼링크를 결정하였으나, 본 시스템에서는 카드가 추가·삭제·변경되면 하이퍼텍스트 색인어가 자동으로 선택되고 마크업되어 하이퍼링크가 자동 설정된다. 이때 하이퍼링크될 색인어와 각 카드 설명문과의 유사도도 자동으로 계산되어 역색인 파일에 등록된다. 설명문과 질의문과의 유사도를 계산할 때 다음과 같은 4가지 고려 사항이 존재한다.

고려사항 1: 질의문에서 중복되는 색인어들을 하나로 취급하는가?

고려사항 2: 설명문에서 중복되는 색인어들을 하나로 취급하는가?

고려사항 3: 질의문의 색인어들은 모두 동일한 가중치를 가지게 할 것인가?

3-1) 색인어들의 위치에 상관없이 1로 둔다

3-2) 색인어가 n개일 때 위치순으로 n에서 1까지 차례대로 가중치를 둔다. 즉, 앞에 발생하는 색인어일수록 높은 가중치를 둔다.

3-3) 3-2로 계산된 가중치에서 전체 가중치의 합으로 나눈 값으로 가중치를 결정한다.

3-4) 사용자로부터 각 색인어들의 가중치를 입력받는다.

고려사항 4: 설명문에서 색인어들은 모두 동일한 가중치를 가지게 할 것인가?

4-1) 위치에 상관없이 모두 동일한 가중치 1을 둔다.

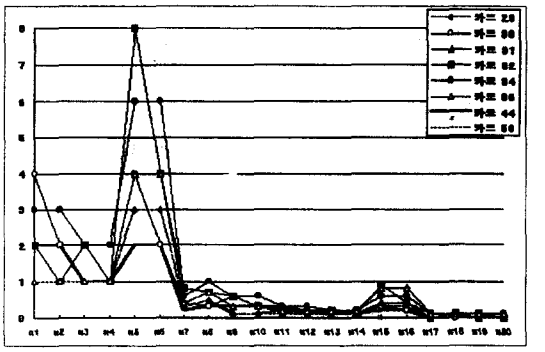
4-2) 1/(전체 색인어의 개수)

〈표 1〉 20가지 유사도 계산 방법들
〈Table 1〉 20 methods of similarity computation

방법 \ 고려사항	1	2	3	4
1	x	x	3-1	4-1
2	x	o	3-1	4-1
3	o	x	3-1	4-1
4	o	o	3-1	4-1
5	o	x	3-2	4-1
6	o	o	3-2	4-1
7	o	x	3-3	4-1
8	o	o	3-3	4-1
9	o	x	3-4	4-1
10	o	o	3-4	4-1
11	x	x	3-1	4-2
12	x	o	3-1	4-2
13	o	x	3-1	4-2
14	o	o	3-1	4-2
15	o	x	3-2	4-2
16	o	o	3-2	4-2
17	o	x	3-3	4-2
18	o	o	3-3	4-2
19	o	x	3-4	4-2
20	o	o	3-4	4-2

(단, o는 yes, x는 no를 뜻함)

위의 네가지 고려사항을 조합한 20가지의 유사도 계산 방법으로 50개의 카드에 대해 실험한 결과를 부록에 실었다. m1~m20은 각각 유사도 계산 방법1~방법20을 뜻하며 각 방법은 〈표 1〉과 같이 네가지 고려사항들을 조합하여 만든 것이다. c1~c50은 카드1~카드50을 뜻하는데 부록의 실험 결과를 자세히 보면 첫 번째 질의문에 대해서는 카드 4~7이, 두 번째 질의문에 대해서는 카드 3~7, 9, 10, 18, 19, 25, 26, 29, 30, 34, 37~40이, 세 번째 질의문에 대해서는 카드 29~34, 36, 44, 50이 그리고 네 번째 질의문에 대해서는 카드 29~32, 34, 36, 38, 39, 44, 50이 후보 카드로 검색됨을 알 수 있다. 이중 질의문 4에 대한 유사도를 정리하여 (그림 7)과 같이 나타내어 보았다. (그림 7)에서 방법 5가 유사도 계산에 있어서 가장 확실한 방법임을 알 수 있다. 다른 질의문에 대해서도 역시 방법 5가 가장 명확히 후보 카드를 찾아 준다.



(그림 7) 20가지 방법으로 측정한 유사도
(Fig. 7) Similarities measured by 20 methods

5. 결 론

본 논문에서는 자주 변화하는 동적 정보 즉, 삽입, 삭제, 변경이 빈번히 일어나는 정보에 대해 색인어를 자동으로 관리하는 방법을 연구하였다. 특히 정보마다 설명문이 첨부되어 있고 이들이 서로 하이퍼링크될 수 있도록 하기 위해, 변경된 부분에 대한 하이퍼텍스트 색인어와 하이퍼텍스트 링크의 변경이 자동으로 이루어지도록 하는 방법을 제시하였다.

종전에는 카드가 삽입, 삭제, 변경될 때마다 사용자

가 직접 변경된 색인어를 찾아 마킹해야 했었지만, 본 논문에서 제시한 방법을 사용하면 동적색인기가 새로이 추가되어야 할 색인어와 제거되어야 할 색인어를 자동으로 발견하여 역색인 파일의 내용을 변화시켜 주므로, 사용자는 색인어의 변화에 신경쓰지 않고 소프트웨어를 개발하고 설명문을 관리할 수 있다. 또 다른 장점은 자연어 질의가 가능하고, 한번 찾은 카드와 연관성이 있는 다른 카드들을 하이퍼텍스트 링크로 검색할 수 있기 때문에, 소프트웨어를 검색의 신축성과 효율을 기대할 수 있다.

본 논문에서 제안한 방법대로 Turbo Pascal for Windows와 Borland resource Workshop, Turbo Debugger for Windows, Whitewater resource toolkit 등의 도구를 사용하여, 객체지향적 방법으로 시스템을 구현하여 실험한 바, 질의문에서는 중복되는 색인어들이 나타나면 이것들을 모두 하나로 취급하고, 역시 질의문의 여러 색인어에 대해서는 위치순으로 앞에서부터 큰 수를 부여하여 1까지 가중치를 주는 방법이 좋은 결과를 낳았다. 그리고 설명문에서는 중복되는 색인어들을 하나로 취급할 필요가 없으며, 설명문의 여러 색인어들의 가중치는 모두 동일하게 1로 주는 것이 후보 카드를 추출하는데 가장 좋은 방법임을 발견하게 되었다.

본 논문에서 사용한 방법의 단점은 소프트웨어 루틴을 추가할 때마다 사용자가 일일이 제목과 설명문을 입력해야 한다는 점과, 문장 분석이 형태소 분석 수준이므로 색인어 추출이 불완전하다는 점이다. 설명문없이 소스 코드만을 대상으로 분류하고 검색하기에는, 기술적으로 어려운 문제들이 남아 있다. 문장 분석에 있어서도 형태소 분석 외에 구문 분석, 의미 분석까지 포함되어야 효과적인 색인어가 추출될 수 있을 것으로 생각되지만, 구문 분석과 의미 분석은 구현하기 매우 복잡하여 실제 정보 시스템에 이용하기에는 아직 이른 것으로 보인다.

본 논문에서 제시한 방법을 개선하기 위해서는 앞서 단점으로 지적했던 설명문 기반 검색을 내용 기반 검색으로 바꾸어야 하며, 보다 효율적인 색인어 추출을 위해서는 구문 분석과 의미 분석이 추가되어야 할 것이다. 그리고 방대한 시소러스를 효과적으로 구축하는 방안, 복합명사를 처리하는 방법 등이 연구되어야 한다.

향후 연구 과제로 분산 환경에서 여러 사람들이 공동작업 하려 할 때 각 사람들이 만든 소프트웨어 루틴을 서로 공유하는 방법을 연구하고자 한다. 자신이 만든 소프트웨어 루틴을 설명서와 함께 저장해 두면, 문장 분석기가 이를 분석하여 어느 부류에 속하는지 조사하고 서버의 데이터베이스에 필요한 정보를 저장한다. 소프트웨어 루틴이 필요한 사람은 자연어로 질의를 하고 서버가 이를 받아 어느 사이트에 있는 어떤 소프트웨어가 가장 적합한 것인지 찾아주는 에이전트를 개발하려고 한다. 유사한 연구가 인터넷 상에서도 가능하다. 본 연구는 이러한 연구의 기초가 될 것으로 기대한다.

참 고 문 헌

[1] 강승식, 권혁일, 김동렬, "한국어 자동 색인을 위한 형태소 분석 기능", 한국정보과학회 봄 학술발표논문집, 1995.

[2] 강승식, "한국어의 형태론적 특성과 형태소 분석 기법", 정보과학회지 12권 8호 1994.

[3] 고영곤, 이택경, 박태진, 최윤철, "하이퍼미디어 시스템과 정보검색", 정보과학회지, 제 13권 제 1호, 1995.

[4] 고영곤, 최윤철, "탐색과 브라우징을 지원하는 하이퍼미디어 시스템의 설계", 정보관리학회지, 제 10권 제 1호, 1993.

[5] 김갑수, 신영길, 우치수, "소프트웨어 부품 자동 추출, 분류 및 검색", 한국정보과학회 봄 학술발표논문집, 제 21권 제 1호, 1994.

[6] 김대진, 정상철, 신동욱, "시소러스를 기반으로 하는 문서순위 결정방법에 관한 연구", 한국정보과학회 봄학술발표논문집, 제 21권 제 1호, pp. 177-180, 1994.

[7] 김명철, 권오욱, 최기선, 김재군, 김영환, "시소러스와 상호정보를 이용한 정보검색모델, 한국정보과학회 봄 학술논문발표집, 1994.

[8] 김민정, 권혁철, "한국어 특성을 이용한 자동 색인 기법", 한국정보과학회 가을학술발표논문집, 제 19권 제 2호, pp. 1005-1008, 1992.

[9] 김정아, 이경환, "객체지향 소프트웨어 개발을 위한 재사용 지원 시스템", 정보과학회논문지(C),

제 1권 제 2호, 1995.

[10] 오길록, 최기선, 박세영, 한글공학, 대영사, pp. 465-502, 1994.

[11] 이영자, 하이퍼텍스트 정보검색에 관한 연구, 도서관학논집 제18집, pp. 91-138, 1991.

[12] 임형묵, 정상철, 신동욱, 김형근, 최기선, "시소러스를 기반으로 하는 자동색인 시스템에 관한 연구", 한국정보과학회 봄학술발표논문집, 제 21권 제 1호, pp. 173-176, 1994.

[13] 정상철, 신동욱, 시소러스 및 요약화일을 이용한 문서 검색시스템, 94'한글 및 한국어 정보처리 학술대회, pp. 400-408, 1994.

[14] 정재헌, 이상구, "정보검색을 위한 효율적인 시소러스 구조에 관한 연구", 한국정보과학괴 봄 학술발표논문집, 1995.

[15] 전국대학 전산학과 연합회, 최신 컴퓨터 대사전, 기다리, 1994.

[16] Eric W. Brown, James P. Callan, W. Bruce Croft, "Fast Incremented Indexing for Full-Text Information Retrieval", Proc. of 20th VLDB Conf. Santiago, Chile, 1994.

[17] Jakob Nielson, Hypertext and Hypermedia, Academic Press, 1990.

[18] Nipon Charoenkikarn, Jim Tam, Mark H. Chignell, Gene Golvchinsky, "Browsing Through Querying: Designing for Electronic Books", Hypertext '93 Proceedings, 1993.

[19] William B. Frakes and Ricardo Baeza-Yates(ed.), Information Retrieval: Data Structures and Algorithms, Prentice-Hall, 1992.

[20] WWW-KR, 가자 Web의 세계로!(개정판), WWW Forum Korea, 1995.

[21] Yorick Wilks, Louise Guthrie, Joe Guthrie, and Jim Cowie, "Combining Weak Methods in Large-Scale Text Processing", in Text-Based Intelligent Systems(edited by Jacobs), Lawrence Erlbaum Associates Pub., 1992.

부 록

20가지의 유사도 계산 방법으로 50개의 카드에 대해 실험한 결과

질의문 1 : 이미지 파일의 이전 이미지를 출력하는 프로그램

가중치 : 이미지 파일 2, 이전 이미지 7

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14	m15	m16	m17	m18	m19	m20
c4	1.0	1.0	1.0	1.0	2.0	2.0	0.7	2.0	0.2	0.2	0.1	0.1	0.1	0.1	0.2	0.2	0.1	0.2	0.0	0.0
c5	2.0	2.0	2.0	2.0	3.0	3.0	1.0	3.0	1.0	1.0	0.3	0.3	0.3	0.3	0.5	0.5	0.2	0.5	0.2	0.2
c6	1.0	1.0	1.0	1.0	2.0	2.0	0.7	2.0	0.2	0.2	0.1	0.1	0.1	0.1	0.3	0.3	0.1	0.3	0.0	0.0
c7	1.0	1.0	1.0	1.0	2.0	2.0	0.7	2.0	0.2	0.2	0.1	0.1	0.1	0.1	0.2	0.2	0.1	0.2	0.0	0.0

질의문 2 : 이미지 파일의 이전 이미지를 출력하는 프로그램

가중치 : 이미지 파일 2, 이전 이미지 5, 출력 2

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14	m15	m16	m17	m18	m19	m20
c3	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
c4	2.0	2.0	2.0	2.0	4.0	4.0	0.7	1.3	0.4	0.4	0.2	0.2	0.2	0.2	0.3	0.3	0.1	0.1	0.0	0.0
c5	3.0	3.0	3.0	3.0	6.0	6.0	1.0	2.0	1.0	1.0	0.3	0.3	0.3	0.3	0.7	0.7	0.1	0.2	0.1	0.1
c6	2.0	2.0	2.0	2.0	4.0	4.0	0.7	1.3	0.4	0.4	0.2	0.2	0.2	0.2	0.4	0.4	0.1	0.1	0.0	0.0
c7	1.0	1.0	1.0	1.0	3.0	3.0	0.5	1.0	0.2	0.2	0.1	0.1	0.1	0.1	0.3	0.3	0.0	0.1	0.0	0.0
c9	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.0	0.1	0.1	0.1
c10	2.0	1.0	2.0	1.0	2.0	1.0	0.3	0.3	0.4	0.2	0.2	0.1	0.2	0.1	0.2	0.1	0.0	0.0	0.0	0.0
c18	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0

c19	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
c25	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
c26	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.0	0.1	0.1	0.1
c29	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.0	0.1	0.0	0.0
c30	2.0	1.0	2.0	1.0	2.0	1.0	0.3	0.3	0.4	0.2	0.1	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0
c34	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
c37	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.0	0.1	0.0	0.0
c38	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
c39	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
c40	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.3	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0

질의문 3 : 인터럽트를 이용하여 드라이브의 정보를 확인하는 인터럽트 프로그램

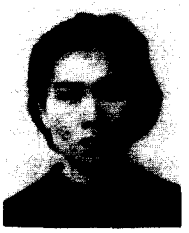
가중치 : 인터럽트 5, 드라이브 5, 정보 2, 인터럽트 5

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14	m15	m16	m17	m18	m19	m20
c29	1.0	1.0	1.0	1.0	3.0	3.0	0.3	0.5	0.1	0.1	0.2	0.2	0.2	0.2	0.8	0.8	0.1	0.1	0.0	0.0
c30	4.0	2.0	2.0	1.0	4.0	2.0	0.4	0.3	0.6	0.3	0.3	0.2	0.2	0.1	0.3	0.2	0.0	0.0	0.0	0.0
c31	1.0	1.0	1.0	1.0	4.0	4.0	0.4	0.7	0.3	0.3	0.1	0.1	0.1	0.1	0.4	0.4	0.0	0.1	0.0	0.0
c32	2.0	1.0	2.0	1.0	8.0	4.0	0.8	0.7	0.6	0.3	0.2	0.1	0.2	0.1	0.9	0.4	0.1	0.1	0.1	0.0
c34	3.0	3.0	2.0	2.0	6.0	6.0	0.6	1.0	0.6	0.6	0.3	0.3	0.2	0.2	0.6	0.6	0.1	0.1	0.1	0.1
c36	1.0	1.0	1.0	1.0	4.0	4.0	0.4	0.7	0.3	0.3	0.2	0.2	0.2	0.2	0.8	0.8	0.1	0.1	0.1	0.1
c44	2.0	2.0	1.0	1.0	2.0	2.0	0.2	0.3	0.3	0.3	0.2	0.2	0.1	0.1	0.2	0.2	0.0	0.0	0.0	0.0
c50	1.0	1.0	1.0	1.0	3.0	3.0	0.3	0.5	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.3	0.0	0.1	0.0	0.0

질의문 4 : 인터럽트를 이용하여 드라이브의 정보를 확인하는 인터럽트 프로그램

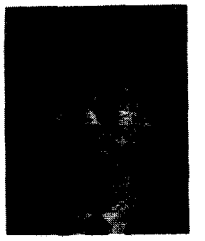
가중치 : 인터럽트 5, 드라이브 5, 정보 2, 확인 2, 인터럽트 5

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14	m15	m16	m17	m18	m19	m20
c29	1.0	1.0	1.0	1.0	3.0	3.0	0.3	0.5	0.1	0.1	0.2	0.2	0.2	0.2	0.8	0.8	0.1	0.1	0.0	0.0
c30	4.0	2.0	2.0	1.0	4.0	2.0	0.4	0.3	0.6	0.3	0.3	0.2	0.2	0.1	0.3	0.2	0.0	0.0	0.0	0.0
c31	1.0	1.0	1.0	1.0	4.0	4.0	0.4	0.7	0.3	0.3	0.1	0.1	0.1	0.1	0.4	0.4	0.0	0.1	0.0	0.0
c32	2.0	1.0	2.0	1.0	8.0	4.0	0.8	0.7	0.6	0.3	0.2	0.1	0.2	0.1	0.9	0.4	0.1	0.1	0.1	0.0
c34	3.0	3.0	2.0	2.0	7.0	7.0	0.5	0.7	0.5	0.5	0.2	0.2	0.1	0.1	0.5	0.5	0.0	0.0	0.0	0.0
c36	1.0	1.0	1.0	1.0	5.0	5.0	0.3	0.5	0.3	0.3	0.2	0.2	0.2	0.2	0.8	0.8	0.1	0.1	0.0	0.0
c38	1.0	1.0	1.0	1.0	3.0	3.0	0.2	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.0	0.0	0.0	0.0
c39	1.0	1.0	1.0	1.0	3.0	3.0	0.2	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.0	0.0	0.0	0.0
c44	2.0	2.0	1.0	1.0	2.0	2.0	0.1	0.2	0.3	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
c50	1.0	1.0	1.0	1.0	4.0	4.0	0.3	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.4	0.4	0.0	0.0	0.0	0.0



이 동 애

1989년 계명대학교 전자계산학과 졸업(학사)
1991년 계명대학교 대학원 전자계산학과(공학석사)
1994년~현재 계명대학교 대학원 전자계산학과 박사과정 수료
1995년~1997년 경북대학교 전산교육센터 전담강사
1997년~현재 영남 전산통계 연구소 부실장
관심분야 : 자연어처리, 정보검색, 인터넷



장 덕 성

1979년 경북대학교 전자공학과 졸업(학사)
1981년 서울대학교 대학원 계산통계학과(이학석사)
1988년 서울대학교 대학원 컴퓨터공학과(공학박사)
1982년~1985년 동아대학교 전산공학과 조교수
1992년~1993년 콜로라도 주립대학 방문교수
1985년~현재 계명대학교 컴퓨터·전자공학부 교수
관심분야 : 자연어처리, 멀티미디어정보검색, 인트라넷