KNetIRS:키워드망을 이용한 정보검색 시스템

우 선 미[†] · 유 춘 식[†] · 이 종 득[††] · 김 용 성[†††]

요 약

기존의 정보검색 시스템들은 질의가 정확하지 않더라도 원하는 정보를 검색할 수 있도록 하기 위해 시소러스(thesaurus)를 사용했다. 그러나 시소러스를 구축하고 유지하는데 드는 비용이 매우 높고 검색에 있어서도 완전하다고 볼 수 없다. 그래서 본 논문에서는 이러한 문제점들을 해결하기 위하여 키워드망을 이용한 정보검색 시스템인 KNetIRS를 설계 및 구현한다.

키워드망은 문서로부터 직접 추출한 키워드들로 구성된다. KNetIRS는 역파일(Iinverted file)의 개념에 기반을 둔 키워드망을 이용하여 데이터베이스에서 적합한 문서만을 탐색한다. 그리고 KNetIRS는 키워드망 브라우저(Keyword Network Browser)를 사용하여 질의를 확장하고, 분할 연산(split function)을 정의하여 "정보 검색", "정보", 그리고 "검색"과 같은 복합어에 관한 처리를 한다.

KNetIRS : Information Retrieval System using Keyword Network

Sun Mi Woo[†] · Chun Sik Yoo[†] · Chong Deuk Lee[††] · Yong Sung Kim[†††]

ABSTRACT

The existing information retrieval systems utilize thesaurus in order to search and retrieve the desired information even when the query is not accurate. However the cost for implementing and maintaining thesaurus is very high and it can not guarantee complete success of search/retrieval operation. Thus in this paper, Information Retrieval System using Keyword Network(KNetIRS) which was designed and implemented to solve these problems is introduced.

Keyword Network composed of keywords which were extracted from documents. KNetIRS finds the appropriate documents by using the Keyword Network which is based on the concept of "Inverted file". In addition, KNetIRS can carry out query expansion by using the Keyword Network Browser, and deal with the conjunction of "정보 검색", "정보", and "검색", by defining and implementing spilt function.

1. 서 론

정보검색이란 방대한 양의 정보를 정보 전문가가 분석, 가공하여 축적하여 둔 매체로부터 사용자의 요구에 적합한 정보만을 찾아내는 일련의 과정을 의미한다[16][17]. 정보를 검색하기 위해 사용되는 기법에는 불리언 논리(Boolean Logic) 검색, 가중치(Weight)에 의한 검색, 매칭함수(Matching Function)에 의한 검색, 클러스터 파일(Cluster File)에 대한 검색, 연관검색(Associative Retrieval), 확률검색(Probabilistic Retrieval), 등이 있다. 본 논문에서 기반을 두고 있는 불리언 논리 검색은 불리언 대수(Boolean algebra)를 이

† 준 회 원:전북대학교 컴퓨터과학과
†† 정 회 원:서남대학교 전산정보학과
††† 종신회원:전북대학교 컴퓨터과학과
논문접수:1997년 2월 11일, 심사완료:1997년 7월 1일

용하여 질의를 만족하는 문헌 집단을 검색한다. 불리언 논리 검색은 컴퓨터를 이용한 처리가 용이하다는 장점이 있으나, 탐색어로 표현되는 각 개념의 중요도를 나타내지 못한다는 단점을 지니고 있다[9][13][16].

하나의 색인어만으로 해당 주제를 모두 검색할 수 없는 경우엔 해당 색인어와 관련된 개념의 용어를 이용하여 검색하면 된다. 그런데 사용자가 관련 개념의 용어를 직접 찾아 질의로 입력하기엔 상당한 불편이 따르게 되므로 기존의 정보검색 시스템들은 시소러스(thesaurus)를 주로 사용하였다[9][15][17]. 그러나 시소러스를 생성하고 관리하는데 드는 비용(cost)이 매우 높고, 질의어와 시소러스 용어 사이의 불일치 문제도 발생할 수 있으며 심지어는 검색 과정에서 원하는 정보를 찾지 못하기도 한다.

따라서 이러한 시소러스의 구축에 관한 여러 문제점들을 해결하고 검색을 보다 효율적으로 하기 위하여 본 논문에서는 키워드망을 이용한 정보검색 시스템(Information Retrieval System using Keyword Network, KNetIRS)을 설계 및 구현한다. KNetIRS는 검색 대상인 논문 제목에서 직접 키워드를 추출하고 그 키워드로 키워드망을 구성하므로 질의와 색인 용어 사이의 불일치 문제를 해결할 수 있다. 그리고 역파일(Inverted file)의 개념을 적용하여 키워드망을 구축함으로써 데이터베이스에서 해당 문서만을 탐색하여 검색 속도를 향상시킨다. 또한 그래픽 사용자 인터페이스(GUI, Graphic User Interface)를 통해 사용자 편의성을 제공하고, 키워드망 관리기 내의 키워드망 브라우저(keyword network browser)를 이용하여 질의 확장을 용이하게 한다. "정보", "검색", "정보 검색"과 같은 복합명사에 관한 문제는 분할(split) 연산을 정의하여 해결하고, 불리언 논리 검색 기법을 기반으로 한 정합 매칭(exact matching)으로 문서를 검색한다.

본 논문의 구성을 보면 먼저 2장에서 본 논문과 관련있는 시소러스, 역파일, 속(Lattice)을 이용한 정보검색 시스템에 대해 간단히 알아본다. 그리고 키워드망의 구성 방법과 이에 필요한 여러 연산들을 3장에서 기술한다. 4장에서는 KNetIRS의 각 구성 요소의 기능과 구현 결과를 기술하고, 마지막으로 5장에서 결론 및 향후 연구 과제에 관해서 논한다.

2. 관련 연구

키워드망을 이용한 정보검색 시스템(Information Retrieval System using Keyword Network, KNetIRS)과 관련있는 시소러스(Thesaurus), 역파일(Inverted file), 속(Lattice)을 이용한 정보검색 방법에 대하여 알아본다.

2.1 시소러스를 이용한 정보검색 시스템

정보시스템과 문헌생산자, 색인작성자, 이용자가 일관성있게 사용할 수 있도록 해당 주제분야에서 필요한 모든 개념을 수집하여 개념들의 대소관계나 동의어, 동형이의어, 관련어 등을 통제하여 둔 용어통제어표를 시소러스라고 한다[17].

이러한 시소러스 내의 용어 관계를 이용하여 검색되는 정보의 양을 조절할 수 있다[28]. 그러나 시소러스의 크기가 커질 경우엔 시소러스의 생성과 관리에 많은 시간이 소비된다는 단점이 있다. 또한 검색시 시소러스 내를 헤맨다거나 용어 사이의 관계를 모두 IS-A 관계로만 취급하는 경우가 발생할 수 있어서 완전한 검색이라고 할 수 없다[18].

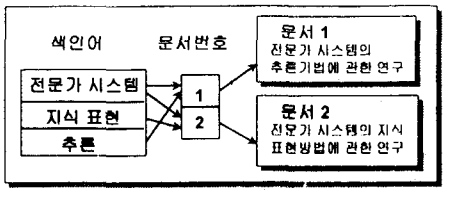
아직까지 모든 학문분야를 망라하고 있는 시소러스는 존재하지 않으며, 주로 주제분야별로 작성되고 있다.

2.2 역파일을 이용한 정보검색

역파일은 기본 파일인 문서 파일과 이에 대한 역색인 파일로 구성되며, 일반적인 파일과 다른 점은 역파일이 주파일의 키필드가 아닌 다른 필드를 키로 하

〈표 2-1〉 문서와 색인어
〈Table 2-1〉 Documents and indices

문서 번호	문서의 색인어
문서 1	전문가 시스템, 추론
문서 2	전문가 시스템, 지식 표현



(그림 2-1) 역파일
(Fig. 2-1) Inverted file

여 색인을 만든다는 점이다[16]. 예를 들어, 문서 화일 내에 〈표 2-1〉과 같이 색인어가 포함되어 있다고 했을 때, 생성되는 역파일은 (그림 2-1)과 같다.

2.3 속을 이용한 정보검색

반순서 집합(partially ordered set)에서 최소 상한(the least upper bound)과 최대 하한(the greatest lower bound)이 존재하면 속이 된다. 속을 이용한 정보검색 시스템은 그래프(헤세도표) 내의 각 정점이 문서에 부여된 색인어들의 곱(conjunction) 형태로 된 질의를 표현하고 있고, 용어 그래프와 문서 부분 집합을 브라우징함으로써 사용자가 점진적인 질의 확장(enlargement) 및 정제(refinement)를 할 수 있도록 하고 있다[5][12].

이 시스템은 사용자 편의성을 제공한다는 장점은 있지만 사용자가 키워드를 직접 입력하지 못하고 단순히 시스템이 제공하는 링크를 따라서만 항해(navigation)할 수 있으며, 문서의 수가 많아지면 속을 구성하기 어렵다[12].

3. 키워드망의 구성

키워드망을 이용한 정보검색 시스템(Information Retrieval System using Keyword Network, KNetIRS)의 성능을 좌우하는 가장 중요한 구성요소인 키워드망(Keyword Network)의 구성 방법에 대해 알아본다.

3.1 키워드

3.1.1 키워드 추출

KNetIRS에서 사용하는 키워드는 논문의 제목에서 명사를 추출하여 사용한다. 키워드가 복합명사일 경우에는 기호 '|'를 사용하여 단일명사 단위로 구분한다. 예를 들면 〈표 3-1〉과 같다.

〈표 3-1〉 키워드 추출
〈Table 3-1〉 Keyword extraction

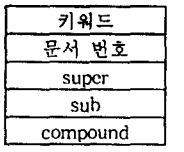
문서번호	논문제목	키워드
7	영상인식을 위한 칼라영상의 영역분할	영상|인식, 칼라|영상, 영역|분할

키워드를 추출한 후 동의어 사전을 이용하여 동의어들을 표준화한다.

3.1.2 키워드 노드의 자료 구조

추출된 키워드를 키워드망에 삽입할 때의 키워드 노드의 자료 구조는 (그림 3-1)과 같이 키워드, 문서 번호, super, sub, compound로 구성된다.

문서 번호는 키워드가 포함되어 있는 문서의 번호를 말하고, super와 sub는 각각 키워드의 상위 개념과 하위 개념을 의미한다. 그리고, compound는 해당 키워드가 포함되어 이루어진 또다른 키워드이거나, 기호 '|'로 나뉘어진 단일명사 키워드를 말한다.



(그림 3-1) 키워드 노드의 자료 구조
(Fig. 3-1) Data structure of keyword node

이때 키워드 노드에 포함된 문서 번호로 직접 데이터베이스의 해당 문서로 접근할 수 있는데, 이 부분이 역파일(Inverted file)의 개념을 응용한 부분이다. 각각의 키워드는 한 개의 super를 가지며 나머지 구성 요소들은 하나 이상의 값을 갖는다. 키워드의 구성 요소들 중에서 키워드명, 문서 번호, super는 추출한 키워드를 키워드망에 삽입할 때 해당 분야의 전문가가 입력하고, 문서 번호를 포함하여 나머지 구성요소는 키워드 삽입시에 본 논문에서 정의한 여러 연산들을 수행하여 시스템이 자동으로 생성한다.

3.2 키워드망 구성 연산

키워드망은 해당 분야의 용어들을 가지고 미리 구성해 놓는 것이 아니라, 검색 대상이 되는 논문에서 추출한 키워드를 키워드망에 추가시킬 때마다 키워드 정보에 따라 계층구조를 형성해간다.

3.2.1 분할 연산

키워드망 내의 키워드들은 논문 제목에서 직접 추출한 키워드와 직접 추출하지 않은 키워드들로 구성

되는데, 전자의 경우에는 키워드가 문서 번호를 갖지만 후자의 경우엔 키워드가 문서 번호를 갖지 않는다. 문서번호를 가진 키워드들만이 분할 연산의 대상이 된다.

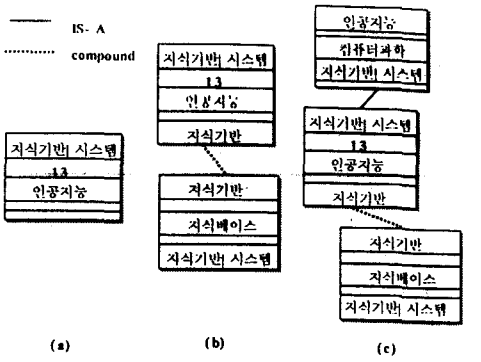
분할 연산 Split_Keyword는 복합명사 키워드, 즉 기호 '|'로 분리된 키워드를 단일명사 키워드로 나누어 각각을 키워드망에 추가시키는 연산으로서 삽입 연산이 수행되는 도중에 수행된다. 그리고 분할 연산 도중에 무결성 검사를 실시하여 키워드 관계성의 모순이나 반복 표기를 방지한다.

분할 연산을 수행하는 목적은 사용자가 해당 키워드의 부분적인 지식만을 가지고도 원하는 문서를 검색할 수 있도록 하는데 있다.

[분할 연산 알고리즘]

```
Split_Keyword(K)
K: 복합명사 키워드, 즉 (A1|..|An)
begin
  for (i=1;i<=n;i++)
  begin
    if (K.Ai가 키워드망에 존재)
      if (Integrity_Check_1(K.Ai) AND
          Integrity_Check_2(K.Ai))
      then K.Ai의 compound 정보에 K를 기록;
    else if (Integrity_Check_2(K.Ai))
    then begin
          K.Ai를 키워드망에 삽입;
          K.Ai의 super를 입력;
          K.Ai의 compound 정보에 K를 기록;
    end
    K의 compound 정보에 K.Ai를 기록;
  end
end.
```

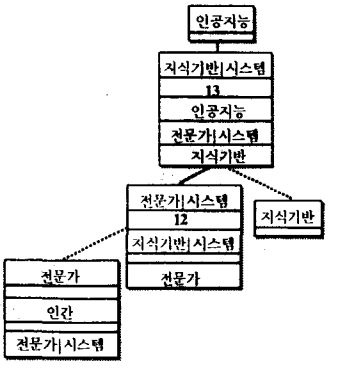
(그림 3-2b)는 (그림 3-2a)와 같은 키워드 "지식기반|시스템"을 키워드망에 삽입하여 분할 연산을 수행한 상태를 나타내고 있다. (그림 3-2a)에서 키워드 "지식기반|시스템"을 포함하고 있는 문서의 번호는 13이고, "지식기반|시스템"의 super는 "인공 지능"으로서 해당 분야의 전문가가 입력한 정보이다. 키워드 "지

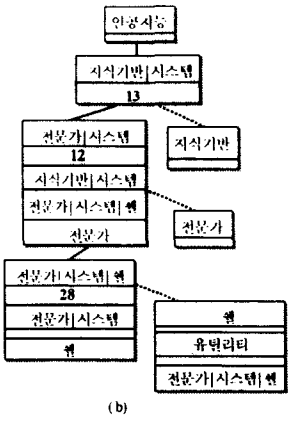


(그림 3-2) 키워드의 분할 ①
(Fig. 3-2) Keyword split ①

식기반|시스템"은 기호 '|'를 기준으로 분할되어 단일명사 "지식기반"과 "시스템"으로 나누어진다. 먼저 단일명사 "지식기반"은 키워드망에 존재하지 않으므로 무결성 검사를 만족하여 compound 키워드로서 키워드망에 삽입된다. 그러나 단일명사 "시스템"은 불용어이므로 무결성 검사를 만족하지 못하여 키워드망에 삽입하지 않는다. (그림 3-2c)는 (그림 3-2b)에 이어서 계층구조 형성 연산이 수행되어 super 키워드인 "인공 지능"을 키워드망에 삽입하고 이 키워드의 super 정보인 "컴퓨터과학"을 전문가가 입력한 상태를 나타내고 있다.

(그림 3-3a)는 (그림 3-2c)에 키워드 "전문가|시스템"을, (그림 3-3b)는 (그림 3-3a)에 키워드 "전문가|시스템|쉘"을 각각 삽입한 후, 분할 연산을 수행한 상태를 나타내고 있다.





(그림 3-3) 키워드 분할 ②
(Fig. 3-3) Keyword split ②

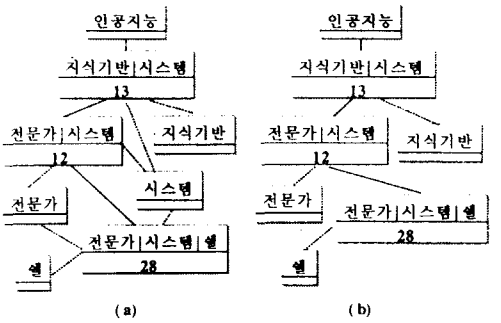
(그림 3-3b)에서 키워드 "전문가|시스템|쉘"은 단일명사 "전문가", "시스템" 그리고 "쉘"로 나누어지지만, 무결성 검사에 의해 키워드 "쉘"만 compound 키워드가 되었다.

3.2.1.1 무결성 검사

무결성 검사(integrity check)는 키워드망을 구축할 때 키워드망의 모순과 반복 표기를 피하기 위해 실시한다.

무결성 검사는 분할 연산 수행 중에 실시하며 제약은 다음과 같다.

1) Integrity_Check_1 : sub는 항상 compound보다 우선한다. 즉, IS-A 관계와 compound 관계가 동일한 경우를 피할 수 있도록 하는 기능이다.



(그림 3-4) 무결성 검사
(Fig. 3-4) Integrity check

2) Integrity_Check_2 : super와 동일하거나 super의 compound와 동일한 compound가 존재하면 super나 super의 compound가 항상 우선한다. 즉, 한 계층에서 compound 관계가 반복되는 경우를 방지하는 기능이다. 또한 불용어이면서 복합명사 분할시에 자동적으로 생성되는 용어도 compound에서 제외한다.

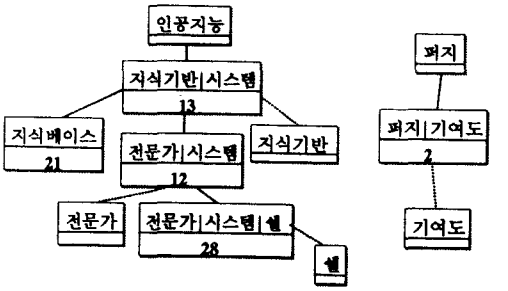
(그림 3-4a)는 분할 연산 수행 중에 무결성 검사를 실시하지 않은 경우의 상태를 나타내고 있고, (그림 3-4b)는 (그림 3-4a)와 같은 상황에서 무결성 검사를 실시한 경우를 나타내고 있다.

3.2.2 삽입 연산

삽입 연산 Insert_Keyword는 키워드망에 새로운 키워드를 삽입시키는 연산으로서, 삽입연산 수행 중에 분할 연산과 계층구조 형성 연산이 수행된다.

[삽입 연산 알고리즘]

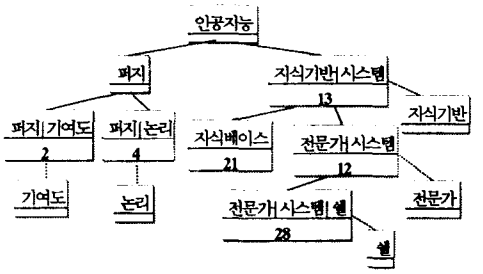
```
Insert_Keyword(K)
  K : 삽입할 키워드
  K′: K와 동일하면서 키워드망에 이미 존재하는 키워드
begin
  if (K가 키워드망에 존재)
     K′에 K의 문서 번호를 추가;
  else begin
     키워드망에 K를 삽입;
     if (K가 복합명사 키워드)
       Split_Keyword(K); /* 분할 연산 수행 */
  end
  Make_Hierarchy(K); /* 계층구조 형성 연산 수행 */
end.
```



(그림 3-5) 키워드 삽입
(Fig. 3-5) Keyword insertion

(그림 3-5)는 (그림 3-3b)에 키워드 "지식베이스"와 "퍼지|기여도"를 삽입한 상태를 나타내고 있다. 키워드 "지식베이스"는 단일명사 키워드이므로 분할 연산을 수행하지 않고 계층구조 형성 연산만을 수행하여 삽입된다. "퍼지|기여도"를 삽입하면 먼저 분할 연산을 수행하여 "기여도"를 compound 키워드로 갖게 되고, super 정보가 "퍼지"이므로 계층구조 형성연산을 수행하여 "퍼지"를 super 키워드로 갖는다.

(그림 3-6)은 (그림 3-5)에 키워드 "퍼지|논리"를 삽입한 상태를 나타내고 있다. 키워드 "퍼지|논리"는 분할 연산에 의해 "논리"를 compound 키워드로 갖고, super 정보가 "퍼지"이므로 계층구조 형성 연산에 의해 키워드 "퍼지"와 연결된다. 그리고 super인 "퍼지"가 키워드망에 존재하고 있으므로 계층구조 형성 연산에 의해 키워드 "퍼지"도 "인공 지능"과 연결된다.



(그림 3-6) 키워드의 삽입 ②
(Fig. 3-6) Keyword insertion ②

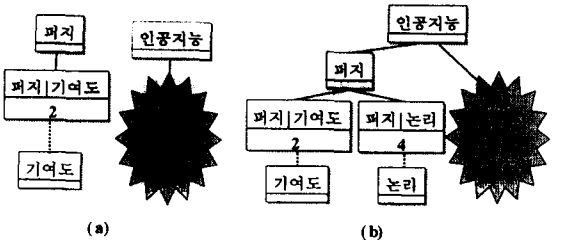
3.2.2.1 계층구조 형성 연산

계층구조 형성 연산 Make_Hierarchy는 키워드망에 키워드를 삽입할 때 super 정보에 관한 처리를 한다. 계층구조 형성은 키워드가 키워드망에 삽입될 때마다, 그 키워드의 super가 키워드망에 존재하는지 존재하지 않은지를 고려하여 super 키워드와 연결을 하거나 새로운 super 키워드를 생성하는 방법으로 이루어진다.

[계층구조 형성 연산 알고리즘]

```
Make_Hierarchy(K)
  K: 삽입되는 키워드
begin
  if (K.super가 키워드망에 존재)
  begin
    키워드망에 존재하고 있는 K. super의 sub 정보에
      K를 기록;
    if (키워드망에 K.super.super가 존재)
    then K.super.super의 sub 정보에 K.super를 기록;
  end
  else begin
      키워드망에 K.super를 삽입;
      삽입된 K.super의 sub 정보에 K를 기록;
      삽입된 K.super의 super 정보를 입력;
  end
end.
```

(그림 3-7b)는 (그림 3-7a)에 키워드 "퍼지|논리"를 삽입하고 분할 연산을 수행한 후 계층구조를 형성한 그림이다. (그림 3-7a)에서 키워드 "퍼지|기여도"는 "퍼지"를 super 정보로 갖고 있는데, 분할 연산 수행 후 계층구조 형성 연산에 의해 super가 키워드망에 있는지를 조사한다. super가 키워드망에 존재하고 있지 않으므로 키워드 "퍼지"를 키워드망에 삽입하고 나서 키워드 "퍼지|기여도"와 super-sub 관계로 연결한다. 그리고 "퍼지"의 super 정보인 "인공 지능"을 해당 전문가가 입력한다.



(그림 3-7) 계층구조 형성
(Fig. 3-7) Hierarchy making

(그림 3-7a)에 키워드 "퍼지|논리"를 삽입하면 "퍼지|논리"의 super 정보가 "퍼지"이므로 분할 연산을 수행한 후, 계층구조 형성 연산에 의해 키워드망에 존재하고 있는 키워드 "퍼지"와 super-sub 관계로 연결된다. 그리고 해당 키워드의 super인 "퍼지"가 이미

키워드망에 존재하고 있는 경우이므로 또다시 "퍼지"의 super 정보가 키워드망에 존재하는지를 조사한다. 키워드 "퍼지"의 super 정보인 "인공 지능"이 키워드망에 존재하므로 키워드 "퍼지"는 키워드 "인공지능"과 super-sub 관계로 연결된다.

3.2.3 삭제 연산

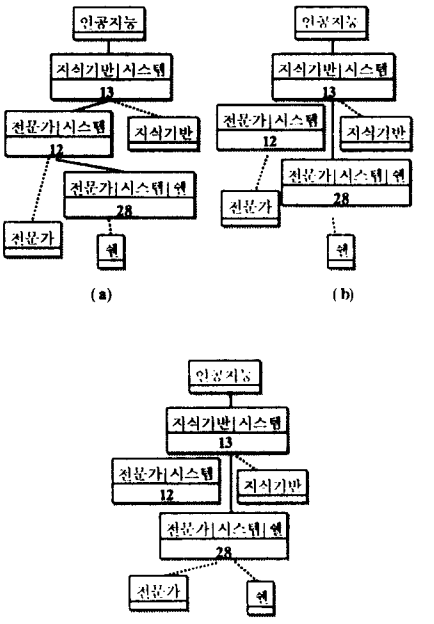
삭제 연산은 두 가지가 있는데, Keyword_Delete_1은 특정 문서를 삭제했을 때 그 문서가 갖고 있는 키워드를 키워드망에서 삭제하는 연산이고, Keyword_Delete_2는 잘못 입력된 키워드이거나 더 이상 필요없게 된 키워드를 키워드망에서 삭제하는 연산이다.

Keyword_Delete_1 연산은 키워드망에 존재하는 해당 키워드의 문서 번호 정보에서 삭제하고자 하는 문서의 번호를 삭제한다. 그리고 Keyword_Delete_2 연산은 키워드망에서 삭제할 키워드에 관한 처리를 한 후, 데이터베이스에서 해당 키워드를 포함하고 있는 문서에 관한 처리를 수행한다.

[삭제 연산 알고리즘]

```
  R : 삭제할 문서
  K : 삭제하고자 하는 키워드
Keyword_Delete_1(K)
begin
  K의 문서 번호 정보에서 R의 문서 번호를 삭제;
end.
Keyword_Delete_2( )
begin
  K.super의 sub 정보에서 K를 삭제;
  if (K.sub가 존재)
  begin
    K.sub의 super 정보를 삭제;
    K.sub의 super 정보에 K.super를 입력;
    K.super의 sub정보에 K.sub를 입력;
  end
  if (K.compound가 존재)
  begin
    K.compound의 compound 정보에서 K를 삭제;
    if ((K.compound의 문서 번호가 NULL) AND
        (K.compound의 sub가 NULL) AND
        (K.compound의 compound가 NULL))
    then K.compound를 키워드망에서 삭제;
    Split_Keyword(K.sub에 존재하는 각 키워드);
  end
  for (K.문서 번호 정보에 존재하는 각 문서들)
  begin
    키워드 정보에서 키워드 K를 삭제;
    if (키워드 정보가 NULL)
    then 데이터베이스에서 문서 레코드 삭제;
  end
  키워드망에서 키워드 K를 삭제;
end.
```

(그림 3-8)은 Keyword_Delete_2 연산의 예로서 키워드 "전문가|시스템"을 삭제하는 과정을 나타내고 있다. 먼저 (그림 3-8a)와 같은 상태에서 키워드 "전문가|시스템"과 연결된 IS-A 관계를 정리하면 (그림 3-8b)와 같이 된다.



(그림 3-8) 키워드의 삭제
(Fig. 3-8) Keyword deletion

(그림 3-8c)는 (그림 3-8b)에 이어 삭제할 키워드의 compound 키워드를 고려한 것으로서 삭제할 키워드와의 연결을 끊은 후, 삭제할 키워드의 sub 키워드에 대해 다시 분할 연산을 수행한다. 이는 분할 연산을 수행했을 때 무결성 검사로 인해 무시되었던 compound 키워드를 고려한 것이다.

마지막으로 키워드 "전문가시스템"을 포함하고 있는 12번 문서를 데이터베이스에서 찾아 그 문서의 키워드 정보에서 "전문가시스템"을 삭제한다.

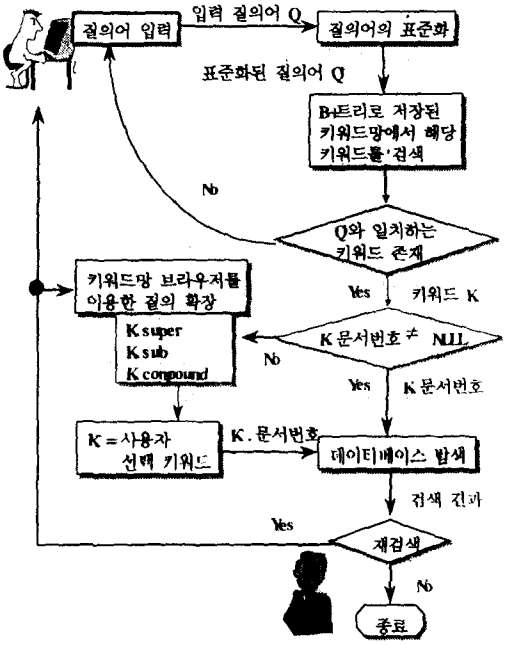
3.3 검색 알고리즘

먼저 사용자가 그래픽 사용자 인터페이스를 통해 질의를 입력하면 검색기가 동의어 사전을 참조하여 질의어의 표현을 표준화시킨다.

표준화된 질의를 가지고 키워드망을 검색하게 되는데, 먼저 질의와 일치하는 키워드를 찾아낸다. 이때 찾아낸 키워드가 문서 번호를 갖고 있을 경우엔 키워드망 검색 과정이 끝나게 되고, 키워드에 문서 번호가 없을 경우엔 키워드망을 재검색하게 된다. 재검색을 할 때, 사용자는 키워드망 브라우저를 이용하여 해당 키워드의 super, sub, compound 키워드를 선택함으로써 질의를 확장할 수 있다. 이러한 과정으로 키워드망 검색을 마치면 해당 키워드의 문서 번호를 결과로 얻게 된다. 키워드망을 검색하여 얻은 문서 번호를 가지고 데이터베이스를 탐색하는데, 문서 번호가 키의 역할을 하게 되어 무작정 데이터베이스를 탐색하는 것보다 탐색 속도를 상당히 줄일 수 있다.

이러한 과정으로 출력된 검색 결과에 만족하면 검색은 완료되고 그렇지 않으면 키워드망 검색 과정으로 되돌아가서 재검색을 하게 된다. 이와 같은 검색 과정을 간략히 그림으로 나타내면 (그림 3-9)과 같다. 시소러스를 이용한 정보검색의 경우에는 시소러스를 이용하여 확장된 질의를 가지고 데이터베이스를 탐색한 후, 검색 결과에 만족하지 않으면 또다시 질의 확장을 거쳐 데이터베이스를 탐색한다. 그러나 KNetIRS에서는 키워드망 내에서 질의를 확장한 후, 그 키워드가 갖고 있는 문서번호를 가지고 데이터베이스 내의 해당 문서로 직접 접근할 수 있다.

데이터베이스 탐색을 마치면 키워드망에서 검색한 키워드를 포함하고 있는 모든 문서들, 즉 논문 제목들이 출력된다. 그리고 출력된 검색 결과들 중에서 하나를 선택하여 보다 자세한 문서 정보 즉, 논문 제목, 저자, 논문지, 페이지, 연도, 키워드에 관한 정보를 볼 수 있다.



(그림 3-9) 키워드망을 이용한 정보검색 알고리즘
(Fig. 3-9) IR algorithm using keyword network

4. 키워드망을 이용한 정보검색 시스템

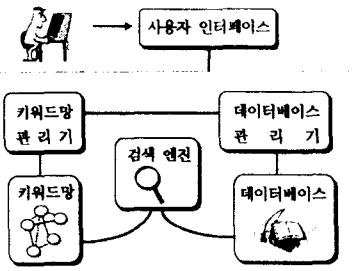
이 장에서는 키워드망을 이용한 정보검색 시스템(Information Retrieval System using Keyword Network, KNetIRS)의 구성과 각 구성요소들의 기능을 상세히 설명한다. 그리고 KNetIRS의 구현 결과에 관해 기술한다.

4.1 KNetIRS의 구성

KNetIRS는 (그림 4-1)과 같이 사용자 인터페이스, 키워드망, 키워드망 관리기, 데이터베이스, 데이터베이스 관리기, 검색기로 구성되어 있으며, SUN상에서 *OSF/Motif toolkit*을 사용하여 구현하였다.

키워드망의 각 키워드 노드들은 키워드명을 키로 하여 자음별로 분류된 다음 B^+-트리로 구성된다. 또한 데이터베이스에는 각 논문에 대한 레코드들이 순

차적으로 저장되어 있으며, 이 레코드들에 대해 저자명을 키필드로 하여 색인파일을 B^+-트리로 구성한다.

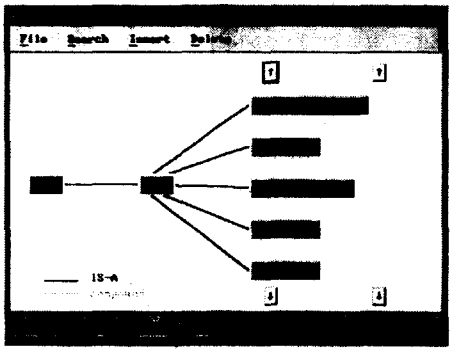


(그림 4-1) KNetIRS의 구성
(Fig. 4-1) Architecture of KNetIRS

사용자 인터페이스는 사용자로부터의 모든 입력과 사용자에 대한 출력을 처리한다. 사용자 인터페이스는 크게 주제선택 윈도우, 정보검색 윈도우, 데이터베이스 관리기, 키워드망 브라우저로 나뉜다. 사용자는 주제 선택 윈도우를 통해 초기에 검색 분야를 선택한다.

4.2 키워드망 관리기

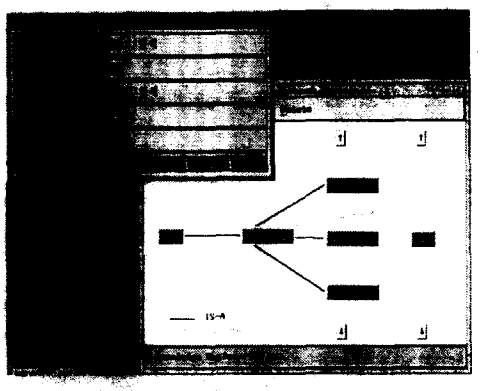
키워드망 관리기는 키워드망에 대한 접근을 용이하도록 하기 위한 키워드망 브라우저를 포함하고 있는데, 이 키워드망 브라우저를 통해 키워드의 삽입, 삭제, 수정이 가능하다. 키워드망 관리기는 이러한 기능들을 수행하기 위해 삽입 연산, 분할 연산 등을 수행한다.



(그림 4-2) 키워드망 브라우저 ①
(Fig. 4-2) Keyword network browser ①

키워드망 브라우저는 사용자가 선택한 키워드를 중심으로 super, sub, compound 관계의 키워드들을 트리 형태로 출력해준다. 한 번에 출력해주는 정보의 개수는 super가 1개, sub와 compound는 최대 5개로 제한한다.

(그림 4-2)는 키워드망 브라우저가 키워드 "패턴"을 중심으로 super, sub, compound 키워드를 출력한 상태를 나타내고 있다. (그림 4-3)은 "search" 기능을 이용하여 키워드 "패턴|인식"을 중심으로 재형성된 키워드들을 보여주고 있다. 그리고, 왼쪽 위 윈도우에서 키워드 "번호|인식"을 선택했을 경우의 정보를 보여주고 있는데, 이러한 구조로 키워드를 삽입하고 수정한다.



(그림 4-3) 키워드망 브라우저 ②
(Fig. 4-3) Keyword network browser ②

해당 키워드의 sub나 compound가 많을 경우에는 키워드망 브라우저의 오른쪽 상단에 위치한 버튼과 오른쪽 하단에 위치한 버튼을 이용하여 모든 관련 키워드들을 볼 수 있다.

4.3 데이터베이스 관리기

데이터베이스 관리기는 문서의 삽입, 삭제, 수정 등 데이터베이스 관리를 담당한다. 또한 데이터베이스 관리기를 통해 저자명을 이용한 문서정보 탐색이 가능하다.

데이터베이스에 저장된 논문은 논문지에서 분류하는 분야별로 구분되어 있다. 그리고 데이터베이스에 저장되어 있는 정보는 문서 번호, 논문 제목, 저자, 논

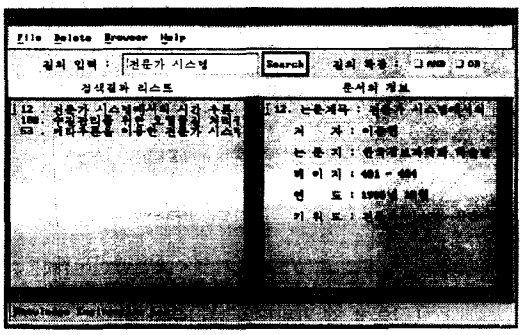
문 발표지, 페이지, 연도, 키워드로 구성되어 있다.

데이터베이스 관리기의 인터페이스인 데이터베이스 관리기는 정보검색 윈도우와 유사하며 저자를 입력하여 원하는 문서를 탐색할 수 있다.

4.4 검색기

검색기는 입력된 질의의 표준화, 키워드망 구성시 키워드의 표준화, 그리고 질의 확장을 수행하며 키워드망과 데이터베이스를 검색한다.

(그림 4-4)는 정보검색 윈도우를 나타내고 있는데, 현재 입력된 질의어가 "전문가 시스템"이고 검색결과 리스트 윈도우에는 입력된 질의어를 포함하고 있는 논문 제목들이 출력되어 있다. 그리고 검색 결과 윈도우에서 선택한 특정 문서에 관한 정보를 문서의 정보 윈도우에서 보여주고 있다.



(그림 4-4) 정보검색 윈도우
(Fig. 4-4) Information retrieval window

검색 결과 리스트 윈도우를 통해서 한 번에 볼 수 없을 정도로 검색 결과가 많으면, 다음과 같은 방법으로 검색 결과를 분류하여 볼 수 있다.

(1) 윈도우 창의 스크롤 기능을 이용하여 본다.
(2) 시스템이 연도, 논문지명과 같은 기준을 제안하고 사용자가 선택하여 결과를 본다.

5. 결론 및 향후 연구

기존의 정보검색 시스템은 질의가 정확하지 않더라도 사용자가 원하는 정보를 찾을 수 있도록 하기 위해 주로 시소러스(thesaurus)를 사용했다. 그러나 시소러스의 구축과 유지에 많은 비용(cost)이 소비되고, 시소러스 안을 헤맨다거나 질의와 시소러스 용어 사이의 불일치 문제가 발생하는 등 검색에 있어서도 완전하다고 볼 수 없다. 그래서 이러한 문제점들을 해결하고 보다 효율적인 검색을 위해 본 논문에서는 키워드망을 이용한 정보검색 시스템인 KNetIRS(Information Retrieval System using Keyword Network)를 설계 및 구현하였다.

KNetIRS는 새로운 환경에서는 키워드망을 재구축해야 하지만 시소러스에 비해 생성과 관리에 소요되는 비용이 매우 낮으며, 역파일의 개념을 응용하여 키워드망을 구성함으로써 검색속도가 향상되었다. 그리고 검색의 대상이 되는 문서에서 키워드들을 직접 추출하여 시소러스의 용어 불일치 문제를 해결하였다. 또한 키워드들을 키워드망에 삽입시킬 때 분할 연산을 수행함으로써 복합 명사에 관한 문제를 해결하였고, 키워드망 브라우저를 이용하여 질의 확장을 효율적으로 하였다.

앞으로 super, sub, compound 이외의 다양한 관계성으로 키워드망을 구축하기 위한 연구와 대량의 정보검색에 필요한 KNetIRS의 효율적인 저장 구조에 관한 연구가 필요하다. 또한 개인의 취향에 맞추어 신속히 정보를 제공하는 적응형 인터페이스에 관한 연구도 계속할 것이다.

참 고 문 헌

[1] Amanda Spink, "Term Relevance Feedback and Query Expansion: Relation to Design," Proc. of the 17th Int'l ACM SIGIR Conference On Research & Development in Information Retrieval, pp. 81-90, 1994.

[2] Augusto Celentano, Silvano Pozzi, et al., "Knowledge based retrieval of office documents," Proc. of the 13th Int'l ACM SIGIR Conference On Research & Development in Information Retrieval, pp. 241-253, 1990.

[3] Brian Vickery, Alina Vickery, "Intelligence and information systems," Journal of Information Science, Vol. 16, pp. 65-70, 1990.

[4] Edward A. FOX, Durgesh Rao, et al., "Users,

User Interfaces, and Objects: Envision, a Digital Library," Journal of the American Society for Information Science, Vol. 44, No. 8, pp. 480-491, 1993.

[5] Gert Schmeltz Pederson, "A Browser for Bibliographic Information Retrieval, Based on an Application of Lattice Theory," Proc. of the 16th Int'l ACM SIGIR Conference On Research & Development in Information Retrieval, pp. 270-279, 1993.

[6] Hatuo Kimoto, Toshiaki Iwadera, "Construction of a Dynamic Thesaurus and Its Use for Associated Information Retrieval," ACM, pp. 227-241, 1990.

[7] James Allan, "Relevance Feedback With Too Much Data," Proc. of the 18th Int'l ACM SIGIR Conference On Research & Development in Information Retrieval, pp. 337-343, 1995.

[8] Kim, Y.W., Kim, J.H, "A model of knowledge based information retrieval with hierarchical concept graph," Journal of Documentation, Vol. 46, No. 2, pp. 113-136, 1990.

[9] Joon Ho Lee, Yoon Joon Lee, et al., "Ranking Documents in Thesaurus-Based Boolean Retrieval Systems," Information Processing & Management, Vol. 30, No. 1, pp. 79-91, 1994.

[10] Neelam Bhalla, "Object-Oriented data models: a perspective and comparative review," Journal of Information Science Vol. 17, No. 3, pp. 145-160, 1991.

[11] Pekka Kilpelainen, Heikki Mannila, "Retrieval from hierarchical texts by partial patterns," Proc. of the 16th Int'l ACM SIGIR Conference On Research & Development in Information Retrieval, pp. 214-222, 1993.

[12] Robert Godin, Jan Gecsel, and Claude Pichet, "Design of a Browsing Interface for Information Retrieval," Proc. of the 12th Int'l ACM SIGIR Conference On Research & Development in Information Retrieval, pp. 32-39, 1989.

[13] Ronald Rousseau, "Extended Boolean Retrieval a Heuristic Approach?," Proc. of the 13th Int'l ACM SIGIR Conference On Research & Development in Information Retrieval, pp. 495-508, 1990.

[14] 전미선, 박세영, "자동 키워드 추출을 위한 키워드망 구축", 한국정보과학회 가을 학술발표논문집, 제21권 2호, pp. 645-648, 1994.

[15] 정영미, "우리말 정보자료를 처리하는 지능형 정보검색 시스템의 설계", 정보관리학회지, 제8권 2호, pp. 4-31, 1991.

[16] 정영미, "정보검색론", pp. 354, 구미무역, 1993.

[17] 이영자, 이경호, "정보학개론", pp. 386, 정각당, 1993.

[18] 박영몽, 김민구, 이정태, "지식기반 정보검색 시스템", 한국정보과학회논문지, 제21권 제11호, pp. 2090-2098, 1994.

우 선 미

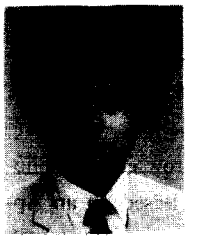
1995년 서남대학교 전자계산학과(이학사)
1997년 전북대학교 대학원 전산통계학과(이학석사)
1997년~현재 전북대학교 대학원 전산통계학과 박사과정
관심분야: 정보검색, 디지털 도서관, 지식공학, 인공지능

유 춘 식

1991년 8월 전북대학교 전산통계학과(이학사)
1994년 전북대학교 대학원 전산통계학과(이학석사)
1994년~현재 전북대학교 대학원 전산통계학과 박사과정
관심분야: 디지털 도서관, 정보검색, 자동색인, 인공지능

이 종 득

1983년 전북대학교 전산통계학과(이학사)
1989년 전북대학교 대학원 전산통계학과(이학석사)
1992년~현재 전북대학교 대학원 전산통계학과 박사과정
1992년~현재 서남대학교 전산정보학과 부교수
관심분야: 멀티미디어 정보검색, 지식공학, 인공지능



김 용 성

1978년 고려대학교 수학과(이학사)
1984년 광운대학교 대학원 전산학과(이학석사)
1992년 광운대학교 대학원 전산학과(이학박사)
1995년 12월~현재 전북대학교 컴퓨터과학과 부교수
1996년 한국정보처리학회 편집위원
1996년 한국정보과학회 학술위원
1996년 12월~현재 한국학술진흥재단 전문위원
관심분야: 디지털 도서관, 정보검색, 지식공학, 멀티미디어, 인공지능