메인 메모리 DBMS를 이용한 정보기술 전문용어 검색 시스템

강옥선* • 경원현** • 조완섭***

An Information Retrieval System for IT Terminologies Using a Main Memory DBMS

요 약

대부분의 일반 정보 검색 시스템은 색인어를 통해 이루어지는데 이런 경우 사용자는 원하는 정보를 얻기 위해 데이터베이스에 저장된 색인어를 정확하게 입력해야 한다. 그러나 일반 사용자가 필요한 색인어를 정확하게 입력하기는 어렵고 특히 원하는 정보가 전문분야의 것일 때는 더욱 그러하다. 따라서 특정 분야의 용어들을 중심으로 전문용어를 관리할 수 있는 시스템의 개발이 요구되고 있다. 정보기술 분야도 빠르게 성장하고 있는 전문분야의 하나로 사용되는 대부분의 단어가 영어이고 한글 표기 또한 다양하여 많은 사용자들이 원하는 정보를 정확하게 찾지 못하고 있다. 이렇듯 단어간의 형태적인 불일치로 인해 생기는 정보 검색의 문제를 해결하고 검색어의 범위를 확장하기 위해 만든 것이 전문용어 검색 시스템이다. 정보 검색시 사용자가 입력한 검색어뿐만 아니라 동의어나 상위어, 하위어까지 검색하여 질의를 확장함으로써 검색 효율을 높일 수 있다. 또한 객체-관계형 데이터베이스로 설계하여 검색이 용이하고, 새로운 단어의 확장이 용이하도록 그 구조를 설계하였다. 제안한 시스템은 메인 메모리 DBMS 를 이용하여 전자상거래와 같이 많은 사용자들이 동시에 접근하는 환경에서도 빠른 검색 성능을 유지할 수 있도록 하였다.

Key words : 정보기술(IT), 전문용어, 시소러스, 정보검색시스템, 메모리 상주 DBMS, 동의어 처리

* 충북대학교 정보산업공학과 (99rain@hanmail.net)
** 충북대학교 정보산업공학과 (space92@trut.chungbuk.ac.kr)
*** 충북대학교 경영정보학과 (wscho@trut.chungbuk.ac.kr)

I. 서론

사용자의 입장에서 볼 때 정보검색 시스템은 방대한 양의 정보들 중에서 요구한 정보를 얼마나 효과적으로 정확하게 찾느냐가 중요하다. 일반적인 정보 검색은 색인어를 통해 이루어지는데 이런 경우 사용자는 정보를 검색하기 위해 데이터베이스에 저장된 정보들이 가지고 있는 색인어를 정확하게 입력해야 한다. 그러나 일반 사용자가 색인어를 정확하게 입력하기는 어렵다. 특히, 찾고자 하는 분야가 전문 분야에서 사용되는 용어일 때는 더욱 그러하다. 이럴 때 시소러스와 같은 지식구조를 이용해서 색인어를 탐색하여 검색의 효율을 높일 수 있다.

최근 들어 정보기술(Information Technology : IT) 분야의 연구가 활발함에 따라 정보자료의 생산이 급속히 증가하고, 이러한 정보자료를 데이터베이스로 구축하여 관련 주제 분야의 연구정보로 활용하고 여러 분야에서 이용하도록 할 수 있는 시스템의 개발이 요구되고 있다. 또한 IT 분야와 같은 전문 분야일 때 검색 시스템에서 활용할 수 있는 용어의 관리에 대한 연구의 필요성이 증가하고 있다. 현재 국내에서는 그에 대한 연구가 미비한 상태이고 특히 정보기술 분야처럼 빠르게 성장하는 분야는 그 필요성이 더욱 커지고 있다.

본 논문에서는 정보기술(Information Technology : IT) 분야에서 정보 검색시에 생기는 용어간의 불일치 문제를 해결하고, 각 용어들간의 계층 관계를 검색하여 정보검색시 검색어의 확장을 도울 수 있는 정보기술 전문용어 검색 시스템 구조를 제안하고 검색 방법을 제안한다. 또한 새로운 단어의 생성이나 삭제, 갱신과 같은 연산이 발생할 때 단어의 계층 구조를 동적으로 확장할 수 있도록 검색 시스템을 구현하였다.

제안된 시스템은 단어간의 계층 구조를 효율적으로 검색하기 위하여 객체-관계형 데이터베이스를 사용하여 구현하였다. 또한 많은 사용자들이 동시에 접근하는 전자상거래와 같은 환경에서도 빠른 검색을 유지할 수 있도록 디스크 기반 DBMS 가 아닌 메인 메모리 DBMS 를 사용하여 검색 시스템을 구현하고 그 비용 모델을 제시한다. 제안된 시스템과 검색 방법은 정보기술 분야뿐 아니라 다른 전문용어 분야로도 그 범위를 확장할 수 있다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 관련연구로서 전문용어 관리시스템과 시소러스, 그리고 메인 메모리 DBMS 에 대해서 살펴본다. 제 3 장에서는 객체-관계형 DBMS 를 이용한 데이터베이스의 구축과 검색 시스템의 구조에 대해 설명한다. 제 4 장에서는 구현된 시스템에서 필요한 연산 알고리즘에 대해 기술하고, 제 5 장에서는 비용을 분석한 후, 제 6 장에서는 결론을 맺고 향후 과제에 대해 기술한다.

II. 관련 연구

이 장에서는 정보기술 분야와 같은 전문 분야에서 사용하는 용어의 관리에 대해 알아보고, 정보검색을 돕는 시소러스의 구조에 대해서 관련 연구들을 살펴본다. 그리고 실제 구현에 사용된 메인 메모리 DBMS 에 대해 살펴본다.

2.1 전문용어 관리 시스템

전문용어는 한 특정 분야의 개념적 정보

와 표현을 용어, 코드, 그래픽 또는 기타 비언어적 기호 및 정의 혹은 다른 서술적 표현을 통하여 나타낸 것을 말한다[Wright & Budin 2001]. 전문용어는 전문화된 정보 및 지식이 사용되는 전 분야에서 중요한 역할을 한다. 정보화가 극도로 진전된 현대사회에서 어느 특정집단이나 개인에 국한된 문제가 아니라 전문용어는 이제 표준화, 정보화, 국제화로 경쟁력을 높이고 산업과 학문의 발전에 기여할 수 있도록 기본 인프라로서 그 중요성이 날로 증대되고 있다.

현재까지 개발된 전문용어 관리 시스템은 약 50 여개가 개발되어 있고, 대부분이 유럽 지역에서 개발되었다. 현재 상이한 접근 방법에 기반한 약 20 개의 전문용어 관리 시스템이 상품화 되어 있다. 국내에서도 물리나 화학, 경제학과 같은 분야에서 연구중에 있다.

2.2 시소러스 (Thesaurus)

시소러스는 문헌정보의 축적과 검색에 있어서 색인 작성자와 검색자가 사용하는 용어를 표준화된 어휘로 통일시킨 용어집으로 서로 개념적으로 관련이 있는 용어들을 미리 주제내용 및 어의체계에 따라 계층관계, 동의어 관계, 관련어 관계를 표시하여 이용자가 빠른 시간내에 정확한 정보를 이용할 수 있도록 체계적으로 배열해 놓은 용어 통제표라고 할 수 있다[ISO 1986]. 정보검색시에 시소러스는 이러한 단어간의 관계성을 이용해 질의에 포함된 용어의 의미를 확대하기 위해 주로 사용된다.

국제적으로 시소러스를 정보검색과 관련하여 사용한 효시는 1959 년에 편찬된 듀퐁(Du Pont)사의 시소러스이고 국내에서는 중앙일보사에서 만든 '중앙 IR-Thesaurus'이다. 이외에도 KEDI 교육 시소러스, 과학기술용어 시소러스, 신문기사 종합 시소러스, 국방과학기술 시소러스 등이 있다.

시소러스에서 사용하는 용어들 간의 관계는 BT(Broader Term:상위어), NT(Narrower Term:하위어), ST(Synonym Term:동의어)와 같은 기호를 사용하고 있다. 동의어란 동일한 개념을 표현하는 두 가지 이상의 용어로서 사실상 대체가 가능하나 시소러스 표준에서는 이 중 하나를 디스크립터(descriptor)로 선택하고 나머지 용어들을 비디스크립터(non-descriptor)로 정한 후 디스크립터만을 색인어로 사용할 것을 권하고 있다. 또한 NT 는 동일 단어그룹 내에서 자신의 위치보다 1 단계 낮은 계층의 단어를 의미하고, BT 는 자신의 위치보다 1 단계 높은 계층의 단어를 의미한다. 시소러스에서 사용하는 기호나 사용법은 시소러스마다 다양하다.

이미 외국의 경우 여러 주제분야에 걸쳐 수백종의 시소러스가 개발되어 자동 색인 및 정보검색시스템 등에 활용되고 있으나 우리 나라에서는 아직 독자적으로 개발된 것은 별로 없고 다만 몇 군데의 전문도서관과 연구소 등에서 외국의 것을 번역하여 그대로 사용하거나 약간 보완시킨 몇 종의 시소러스가 있는 실정이다.

2.3 메인 메모리 DBMS

현재 널리 사용되고 있는 디스크 기반 데이터베이스 시스템(Disk resident database system)은 데이터를 다루기 위한 디스크 액세스의 오버헤드가 지나치게 크므로 빠른 처리 속도를 요구하는 응용에 적합하지 않다[황규영 1996]. 이에 비해 주기억장치 DBMS 는 빠른 접근 시간과 균일한 성

능 분포 특성으로 인하여 실시간 고성능을 요하는 응용 분야에서 각광을 받고 있으며 [Amman 1985, DeWitt 1984], 여타 분야에서도 데이터베이스 응용 프로그램의 응답시간을 단축시켜 더 많은 서비스를 제공하도록 그 응용 분야가 점차 확대되어 가고 있다. 따라서 본 연구에서 제안하는 검색 시스템과 같은 구조에도 적합하다고 할 수 있다.

III. IT 전문용어 검색 시스템

3.1 시스템 개발 환경

제안한 전문용어 검색 시스템은 IT 분야의 업체정보나 제품정보, 기술정보, 그리고 전문가 정보 등을 검색할 수 있는 IT 정보 검색시스템에서 IT 분야의 전문용어를 검색하기 위한 것으로 정보 검색시 사용자가 정확한 검색어를 입력하지 않더라도 검색어를 확장하여 검색 효율을 높이기 위해 구현하였다. 이를 위해 사용자가 입력한 검색어에 대해 동의어나 상위어, 하위어를 검색하고 관리하기 위한 모듈과 질의를 확장하기 위한 모듈로 구성되어 있다.

본 논문에서 전문용어라 함은 정보기술 분야의 용어를 의미하고 단어들은 실제 업계나 연구분야에서 많이 쓰이는 단어들을 중심으로 관련 서적과 전문가들의 도움을 받아 수집하였다. 수집된 단어들은 다시 주제별로 나누어서 키워드별로 정리했다. 정리된 단어들은 그 계층 관계를 표현하기 위하여 트리 형태로 재구성하였다. 트리 구조에서 노드는 디스크립터이고 간선은 단어들 간의 관계를 표현하며 비디스크립터는 트리 구조에는 나타나지 않는다. 트리로 표현된 단어들은 데이터베이스에 저장하여 용어 데이터베이스를 구성하였다.

제안한 검색 시스템과 방법은 실제 IT 정보검색시스템에 포함되어 검색 효율을 높이기 위해 사용된다.

본 연구에서는 제안한 IT 검색 시스템은 IT 분야의 동적인 변화에 신속 정확하게 대응할 수 있도록 다음과 같이 설계하였다.

첫째, 기존에 많이 사용하던 관계형 데이터베이스가 아닌 객체 기반 구조로 설계하여 모든 단어와 계층을 하나의 객체로 인식하게 하였다.

둘째, 포인터를 두어 빠른 검색을 지원할 수 있게 하였다.

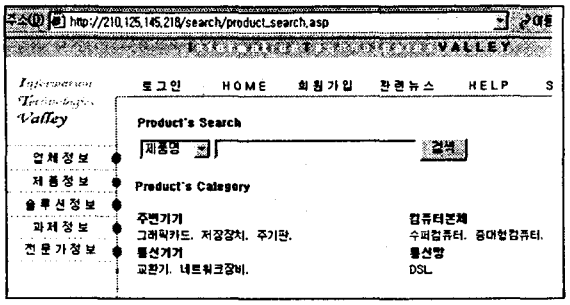
셋째, 일반 디스크 기반의 DBMS 가 아니라 메인 메모리 DBMS 를 사용하여 성능의 향상을 가져왔다. 메인 메모리 DBMS 는 다른 디스크 기반의 DBMS 에 비해 디스크의 I/O 를 없애 속도면에서 빠른 성능을 보인다. 따라서 전자상거래나 본 논문에서 구현한 정보검색 시스템과 같이 많은 사용자들이 빈번하게 이용하는 시스템에서 우수한 성능을 보일 것으로 기대된다.

마지막으로, GUI 를 이용한 사용자 환경을 제공하여 일반 사용자가 손쉽게 이용할 수 있도록 하였다.

이를 위해 DBMS 는 국내의 전자통신 연구원에서 개발된 메인 메모리 객체-관계형 DBMS 인 "Tachyon"[ETRI 2000]을 이용하였다.

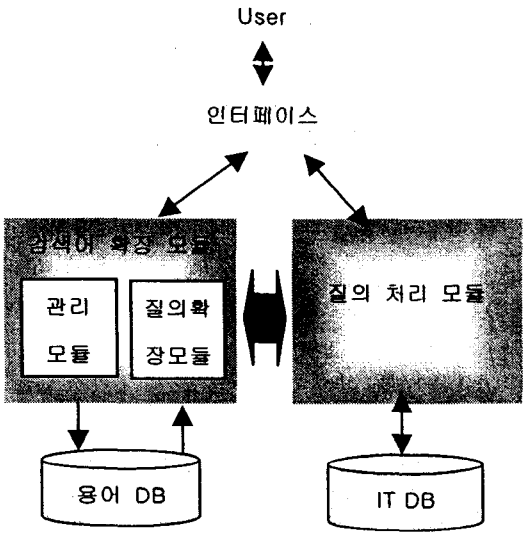
3.2 검색 시스템의 구조

전체 검색 시스템은 IT 업체나, 제품, 기술정보, 용역과제, 전문가정보 등의 IT 정보를 검색하기 위한 시스템으로 현재 개발중에 있다. [그림-1]은 IT 정보 검색 시스템에서 제품 정보를 검색하기 위한 화면이다.



[그림-1] IT 검색 시스템의 검색 윈도우

이때 검색 시스템에서 사용되는 검색어를 관리하기 위한 용어관리 시스템의 구조가 [그림-2]에 나타나 있다.



[그림-2] IT 정보검색시스템

검색 시스템의 구조는 사용자 인터페이스, 단어의 검색과 관리, 질의 확장을 위한 검색어 확장 모듈, 그리고 실제 IT 정보를 검색하기 위한 질의 처리 모듈로 구성된다.

3.2.1 사용자 인터페이스

사용자 인터페이스는 그래픽 사용자 인터페이스(GUI) 환경으로 사용자가 검색할 단어를 입력하고 검색하는 부분이고 이를 통해 검색 결과를 확인할 수 있다.

3.2.2 검색어 확장 모듈

검색어 확장 모듈은 단어의 검색 및 관리를 위한 관리 모듈과 검색된 단어들로 새로운 질의를 형성하는 질의 확장 모듈로 나누어진다. 일단 사용자 인터페이스를 통해 들어온 검색어는 관리 모듈을 통해 용어 데이터베이스에서 적합한 색인어와 동의어, 상위어, 하위어를 검색한다. 관리 모듈에서는 또한 새로운 단어의 삽입과 삭제, 갱신과 같은 작업을 한다.

관리 모듈을 통해 검색된 색인어와 동의어, 상위어, 하위어들은 질의확장 모듈로 보내어져 질의 변환하여 새로운 질의를 구성함으로써 검색 효율을 높일 수 있다.

3.2.3 질의 모듈

질의 확장 모듈에서 변환된 질의는 질의 모듈에서 실제 IT DB 에 접근하여 정보를 검색한다.

3.3 데이터 베이스 구조

데이터베이스는 전문용어를 저장하기 위한 용어 데이터베이스(용어 DB)와 업체나 기술, 제품 정보 등에 대해 저장한 IT 데이터베이스(IT DB)로 구성되어 있다. 그 중 용어 데이터베이스는 용어 클래스(Term Class)와 계층 클래스(Ht Class) 두 개의 클래스로 구성되어 있다.

3.3.1 용어 클래스 (Term Class)

Term 클래스는 IT 분야의 모든 전문용어를 등록하는 클래스이다. 구조는 [그림-3]와 같다. Term 클래스에 있는 단어는 고유(unique)해야 하며 반드시 존재해야(not null) 한다.

WORD	WORD_REF
데이터베이스	@10|201
정보산업	@10|202
DB	@10|201
Database	@10|201
:	:

[그림-3] Term Class

[그림-3]에서 WORD 애트리뷰트는 디스크립터와 비디스크립터를 모두 포함하고 있으며, WORD_REF 애트리뷰트는 Ht 클래스의 객체 식별자(Object Identifier :OID)를 속성으로 가짐으로써 Ht 클래스를 참조하고 있다. 검색시 Term 클래스를 가지고 단어간의 관계를 검색할 때 Ht 클래스와 조인이 요구된다. 이때 WORD_REF 에 있는 OID 를 사용함으로써 객체를 직접 접근하여 조인 비용을 줄일 수 있다. 예를 들어 사용자가 'database'라는 검색어를 사용한다면 기존의 검색 시스템은 '데이터베이스'나 'DB'는 검색할 수 없지만 제안한 구조에서는 모두 계층 클래스의 같은 OID 를 참조함으로써 계층 클래스에 직접 접근하여 관계성을 검색하여 질의를 확장할 수 있다.

3.3.2 계층 클래스 (Ht Class)

Ht 클래스는 단어의 계층 관계를 보여주는 클래스로 디스크립터만으로 구성되어 있다. [그림-4]에서 보는 것과 같이 계층 클래스는 색인어(keyterm), 동의어의 집합(ST), 상위어(BT), 하위어의 집합(NT)으로 구성된다. Tachyon 객체-관계형 DBMS 는 애트리뷰트에 집합값을 가질 수 있어서 동의어나 하위어의 경우 집합값을 허용한다.

KEYTERM	ST	BT	NT
데이터베이스	DB,database	정보산업	RDB, …
정보산업		컴퓨터	데이터베이스, …
컴퓨터	Computer		하드웨어, …
:	:	:	:

[그림-4] Ht Class

IV. 모듈별 연산

4 장에서는 관리모듈과 질의확장모듈에서 사용되는 알고리즘에 대해 설명한다. 사용자가 입력한 검색어를 찾기 위한 검색 알고리즘과 새로운 단어가 생성되었을 때 입력하기 위한 삽입 알고리즘, 그리고 삭제되었을 때 발생하는 삭제 알고리즘에 대해 설명한다. 그리고 검색된 단어들을 이용해 질의를 변환하기 위한 질의확장 알고리즘에 대해 설명한다.

4.1 검색 알고리즘

사용자가 입력한 검색어는 곧바로 IT DB 에서 검색되는 것이 아니라 관리 모듈을 통해 용어 DB 를 검색하여 동의어 및 상위어, 하위어를 검색하게 된다. 이렇게 검색된 단어들은 검색의 효율을 높이기 위해 질의를 확장하는데 사용된다. 검색 연산에 사용되는 검색 알고리즘은 다음과 같다.

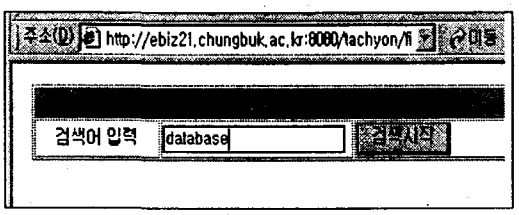
Search Algorithm ( )

input : 검색어 'K'
Output : 키인덱스, 동의어 집합, 상위어, 하위어집합

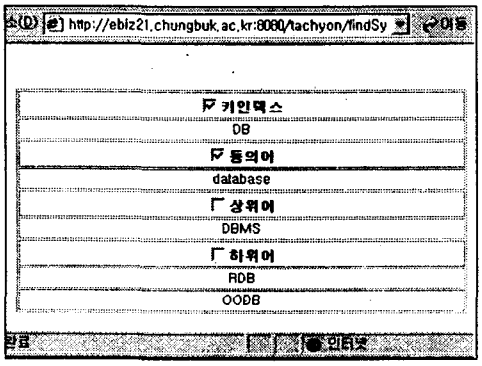
step1 : Term class 에서 검색어 K 를 찾는다.
step2 : 찾아진 K 의 포인터를 따라 ht class 로 이동한다.
step3 : ht class 에서 키인덱스, 동의어집합, 상위어, 하위어 집합을 리턴한다.

사용자가 검색어 K 를 입력하면 일단 Term 클래스에서 단어를 검색한다. 찾아진 검색어가 있다면 OID 를 참조하는 포인터를 따라 Ht 클래스로 이동하여 포인터가 가리

키는 인스턴스를 찾아서 키인덱스와 동의어 집합, 상위어, 하위어 집합을 돌려 받는다. [그림-4]는 검색을 위한 사용자 인터페이스이고 [그림-5]는 검색 결과이다.



[그림-4] 검색 윈도우



[그림-5] 검색 결과 윈도우

[그림-5]에서 보는 것처럼 키인덱스와 동의어는 기본적인 확장에 사용되지만 상위어나 하위어는 사용자가 원할 경우 확장 할 수 있도록 하였다.

4.2 삽입 알고리즘

새로운 단어가 생성되면 그에 대한 동의어나 상위어, 하위어 등의 계층 관계가 발생한다. 따라서 그러한 관계들을 모두 포함하여 단어를 삽입해 주어야 한다. 삽입 알고리즘은 다음과 같다.

Insert Algorithm ( )

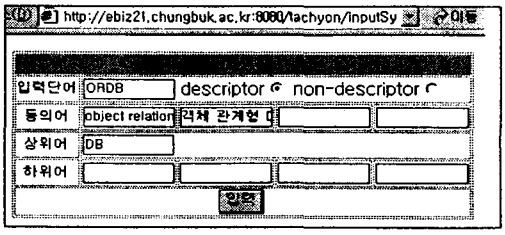
Input
K : 삽입할 단어
SS : K의 동의어 집합
B : K의 상위어
NS : K의 하위어 집합

setp1 : K가 디스크립터일 때
① Ht class에 K를 삽입하고 계층관계를 표시한다.
② Term class에 K와 동의어 집합 SS'를 각각 삽입한다.
③ K의 상위어 B가 있다면 ht class에서 B의 하위어 집합을 갱신한다.
④ K의 하위어 NS가 있다면 ①~③의 과정을 반복한다.

step2 : K가 비디스크립터일 때,
① Term class에 'K'를 삽입한다.
② Ht class에서 'K'의 동의어 즉, 디스크립터인 'M'의 동의어 집합을 갱신한다.

삽입 알고리즘에서 K는 삽입될 단어로 먼저 디스크립터인지 비디스크립터인지를 구별해 주어야 한다. 만약 K가 디스크립터인 경우 K는 Term 클래스와 Ht 클래스에 모두 삽입해 주어야 하며 비디스크립터인 경우에는 Term 클래스에만 삽입하면 된다. K가 디스크립터이고 K와 동의어 관계에 있는 용어들(SS)이 있다면 동의어들도 Term 클래스에 삽입해 준다. 그리고 상위어가 존재한다면 Ht 클래스에서 상위어(B)를 갱신해 준다. 만약 하위어(NS)가 있다면 하위어도 하나의 디스크립터로 취급하여 단어의 삽입과정을 반복한다.

[그림-6]은 삽입을 위한 인터페이스이다.



[그림-6] 삽입 윈도우

4.3 삭제 알고리즘

특정 단어가 삭제되는 것은 좀 더 복잡하게 생각해야 한다. 디스크립터와 비디스크립터인 경우를 구별해야 함은 물론이고, 디스크립터인 경우 동의어는 같이 삭제되는 것으로 간주한다. 그러나 하위어가 있다면 문제가 된다. 이때는 구조를 재구성하기로 한다. 삭제 알고리즘은 다음과 같다.

Delete Algorithm( )

Input
K : 삭제할 단어
M : K 의 디스크립터
B : K 의 상위어

step1 : K 가 디스크립터일 때,
① Term class 에서 K 와 동의어 집합 'SS'를 삭제한다.
② Ht class 에서 K 를 삭제한다.
③ K 의 상위어가 있다면 상위어 B 의 하위어 집합을 갱신한다.

step2 : K 가 비디스크립터일 때,
① Term class 에서 K 를 삭제한다.
② Ht class 에서 K 의 디스크립터 M 의 동의어 집합을 갱신한다.

삭제 알고리즘도 삽입과 마찬가지로 삭제될 단어가 디스크립터인지 비디스크립터인지를 구별해 주어야 한다. 만약 삭제될 단어 K 가 디스크립터라면 Term 클래스와 Ht 클래스에서 모두 K 를 삭제해 주고 K 의 동의어들도 Term 클래스에서 삭제한다. 또한 Ht 클래스에 K 의 상위어 B 가 있다면 B 의 하위어에서 K 를 삭제해 준다. 만약 비디스크립터라면 Term 클래스에서만 삭제하고, Ht 클래스에서 동의어 관계에 있는 키인덱스 M 의 동의어 집합에서 K 를 삭제해 준다.

4.4 질의 확장 알고리즘

관리 모듈에서 찾아진 검색어들의 집합은 질의 확장 모듈을 통해 질의 변환을 하게 된다. 질의 확장 모듈은 검색어를 확장함으로써 검색 효율을 높이고자 하는 모듈이다. 만약 사용자가 검색어를 'database'라고 입력하더라도 관리 모듈에서 찾아진 'DB, database,데이터베이스'를 모두 검색어의 범위에 넣어 검색어의 범위를 확장하는 것이다. 질의 확장 알고리즘은 다음과 같다.

Query Reformulation Algorithm ( )

input : keyterm, st, bt, nt
output : rewritten query Q'

step1 : 관리모듈에서 검색된 키인덱스, 동의어집합, 상위어, 하위어 집합을 받는다.
Setp2 : source query 의 where 절에 검색된 단어들을 OR 조건으로 추가하여 새로운 질의 Q' 을 생성한다.

질의 확장 알고리즘에서 입력 값은 검색 모듈에서 찾아진 검색어들의 집합이다. 원시 질의가 사용자가 입력한 검색어만을 가지고 검색하는 질의라면 질의 확장 모듈을 거친 재구성된 질의는 검색 모듈에서 찾아진 모든 검색어들을 포함하는 질의이다.

예를 들어, 사용자가 DATABASE 와 관련된 업체의 홈페이지 URL 을 찾고자 할 때, 사용자는 검색어를 'database' 라고 입력할 것이다. 이때 원래의 질의를 Q 라고 한다면 Q 는,

```
SELECT    url
FROM      T1
WHERE     index = 'database'  ;
```

가 된다. 이를 질의 확장 모듈을 통하게 되

면 새로운 질의 Q'은,

```
SELECT   url
FROM     T1
WHERE    index in('데이터베이스',
                  'DB','database') ;
```

가 된다. 따라서 이전 질의 Q 에서 검색하지 못한 부분까지도 검색하여 검색 효율을 높일 수 있다.

V. 비용 분석

본 논문에서는 기존의 디스크 기반 관계형 DBMS 에서의 비용과 제안한 메모리 상주 객체-관계형 DBMS 에서의 비용을 저장 공간 비용과 질의 처리 비용의 두 가지 측면에서 비교하고자 한다.

저장 공간은 실제 데이터가 저장되어 있는 테이블이 차지하는 공간과 인덱스가 차지하는 공간 비용으로 계산 할 수 있다. 관계형 DB 에서는 일반적으로 많이 사용하는 B-트리 인덱스를 가정하고, 객체-관계형 DB 에서는 메인 메모리 DB 에서 많이 사용하는 T-트리 인덱스를 가정하였다. 저장 공간비용은 다음의 식(1)과 같이 계산할 수 있다.

$$(1)\ SC = TS + IS$$
$$= \left[p(Term) + p(Ht) + p(St)\right] + IS\ pages$$

SC : 저장공간비용(storage cost)

TS : 테이블 저장 공간

IS : 인덱스 저장 공간

P(X) : X 테이블을 저장하는데 필요한 페이지 수

인덱스의 저장 공간 비용을 보았을 때 B-트리와 T-트리 인덱스의 저장 공간의 차이가 그리 크지는 않으나 T-트리가 약간 우수한 것으로 나타나 있다[Lehman 1986]. 실제로 제안한 구조에서는 동의어나 계층 테이블을 분리하지 않고 하나의 클래스에서 처리할 수 있어서 객체-관계형 DB 를 이용했을 때 약 8%정도의 공간 비용상의 이득을 얻을 수 있다.

질의 평가의 비용은 디스크 액세스 비용, 질의 처리를 위한 CPU 시간 및 분산 또는 병렬 데이터베이스 시스템에서의 통신 비용 등과 같은 기준으로 측정할 수 있다. 여기서 통신 비용은 모두 같다고 가정하더라도 비용 분석은 디스크 기반과 메모리 기반에서 차이가 있다. 대부분의 디스크 기반 질의 처리 비용의 평가에서는 디스크 액세스 비용이 메모리 내 연산보다 많은 비용을 차지하므로 디스크 액세스 비용을 기반으로 평가해왔다. 그러나 메모리 상주 DBMS 에서는 디스크 액세스가 일어나지 않으므로 여기에서는 원하는 데이터를 검색하는 필요한 디스크 액세스 비용과 CPU 시간의 두 가지를 모두 비교하고자 한다.

디스크 기반 RDB 에서 동의어를 검색하기 위한 질의 처리 비용은 다음과 같다.

$$(2)\ COST(RDB)$$
$$= W * \left|page\_read\right| + \left|comparison\right|$$
$$= W * \left[(\log N(Term) + 1) + (\log N(St) + 1)\right]$$

|page_read| : 질의를 실행하기 위해 읽어야 할 페이지 수

|comparison| : 원하는 데이터를 찾기 위해 비교해야 할 레코드 수

W : 비교연산에 대한 디스크 액세스 비용의 상대적 크기

N(Term) : 용어 테이블의 튜플 수

N(St) : 동의어 테이블의 튜플 수

식(2)에서 |page_read|는 질의처리를 위해 읽어야 할 페이지 수로 디스크 액세스를 의미하고, |comparison|은 원하는 레코드를 찾기 위해 비교해야 할 레코드 수 즉, CPU 시간을 의미한다. 식(2)는 동의어만을 검색하기 위한 비용이고, 실제로 상위어나 하위어의 계층 관계를 모두 검색하기 위해서는 계층 테이블을 검색하는 비용이 더 필요하다. 그리고 검색은 모두 인덱스를 이용한다고 가정할 때의 비용으로 원하는 단어를 찾기 위해 인덱스를 이용하여 용어 테이블에서 단어를 찾은 후, 동의어 테이블과 조인하는 비용을 더한 것이다. W 는 메모리 내에서의 연산을 1 로 보았을 때 디스크 액세스 비용의 상대적인 크기로 보통 10 에서 30 의 범위를 갖는다[DeWitt 1984].

메모리 상주 ORDB 에서의 질의 처리 비용은 디스크 I/O 가 일어나지 않으므로 CPU 시간만으로 비용을 계산할 수 있다. 또한 두 개의 테이블을 조인하기 위한 연산이 없이 단지 OID 의 참조관계에 의해 암시적 조인이 발생하므로 조인 비용을 줄여 비용상의 이득을 볼 수가 있다. ORDB 를 이용한 비용은 다음과 같다. 여기서도 마찬가지로 용어 클래스에는 T-트리 인덱스가 있다는 것을 가정한다.

$$(3)\ COST(ORDB) = |comparison| = \lceil (\log N(Term)) \rceil + 1$$

N(Term) : 용어 클래스의 인스턴스 수

식(3)에서 log N(Term)은 용어 클래스에 T-트리 인덱스가 있다고 가정했을 때 원하는 데이터를 찾기 위한 비용이고 '1'은 찾은 인스턴스의 객체 OID 를 참조하는 비용이다. 관계형 DB 에서 필요한 조인 비용을 줄일 수 있다는 것을 보여준다.

위의 비용 식으로 보아 메모리 상주 객체-관계형 DBMS 를 사용하였을 때는 질의처리 비용에서 큰 오버헤드를 차지하는 디스크 액세스 비용을 없앨 수 있고, 또한 객체-관계형 데이터베이스를 사용함으로써 여러 테이블간의 조인 질의를 없앨 수 있어 빠른 검색 성능을 얻을 것으로 보인다. 실제로 식(2)와 식(3)을 비교해보면, 디스크 액세스의 상대적 크기를 20 으로 볼 때 메인 메모리 객체-관계형 DBMS 를 사용했을 때는 대략 20 배 이상의 성능 차이가 있을 것으로 보인다.

VI. 결 론

정보기술 분야의 산업 및 연구가 늘어나고 그에 관련된 정보를 이용하는 분야가 늘어남에 따라 IT 정보를 이용하고 관리할 수 있는 시스템이 요구되고 있다. 이에 IT 정보 검색 시스템을 개발하고, 검색시스템에서 사용할 수 있는 용어의 관리를 위한 전문용어관리 시스템을 개발하였다.

본 논문에서 제안한 검색 시스템은 정보기술 분야에서 사용되는 용어를 중심으로 검색어의 확장을 돕기 위한 것으로 다음과 같은 잇점이 있다.

첫째, 정보 검색시에 사용자가 색인어를 정확히 알지 못하더라도 제안한 시스템은 색인어의 계층 구조를 효과적으로 탐색하여 기존의 검색시스템에서 찾지 못했던 부분까지 검색함으로써 검색 효율을 높일 수 있다.

둘째, 새로운 단어의 삽입과 삭제시에도 단어의 계층 구조의 동적인 변화가 가능하

도록 설계하였다.

셋째, 구현한 데이터베이스 구조와 검색 방법은 IT 분야뿐 아니라 다른 여러 응용 분야에도 활용될 수 있다.

마지막으로 여러 명의 사용자가 동시에 접근하는 전자상거래와 같은 환경에서도 빠른 검색 성능을 유지 할 수가 있다. 이를 위해 기존의 디스크 기반 DBMS 가 아니라 메모리 상주 DBMS 를 이용하여 디스크 액세스를 없애고, 객체-관계형 데이터베이스를 사용하여 테이블간의 조인이 아닌 객체 식별자를 참조함으로써 조인 없이 직접 접근하여 성능향상을 가져왔다.

향후 제안한 시스템의 성능을 실제로 평가해보고, 구현한 검색 시스템은 실제 응용 시스템에서 그 실용성을 평가하여 검색 시스템으로서의 정확성을 평가해야 하겠다. 또한 다른 여러 전문분야나 검색 시스템에도 활용하는 방안이 필요하다.

참 고 문 헌

정재헌, 이상구, "정보 검색을 위한 효율적인 시소러스 구조에 관한 연구", 정보과학회지, 1995.

황규영, "주기억장치 데이터베이스를 위한 저장 시스템", 1996.

A.Ammann, M.Hanrahan, and R.Krishnamurthy, "Design of a Memory Resident DBMS," Proc. Intl. Conf. on COMPCON, 1985.

D.J.DeWitt, R.H.Katz, F.Olken, "Implementation Techniques for Main Memory Database Systems," Proc. Intl. Conf. on management of Data, ACM SIGMOD, pp.1-8, 1984.

H.Garcia-Molina and K.Salem, "Main Memory Database Systems: An Overview," IEEE Trans. on Knowledge and Data Engineering, Vol.4, No.6, pp.509-516, 1992.

S.Gauch, J.B.Smith, "Search Improvement via Automatic Query Reformulation," ACM Trans. on Information Systems, pp.249-280, 1991.

International Organization for Standardization, ISO 2788 : Guideline for the Establishment and Development of Monolingual Thesauri, 2nd ed., 1986.

William B.Frakes, Information Retrieval, Prentice Hall, 1992.

T.Lehman and M.Carey, "A Study of index Structures for Main Memory Database Management System," Proc. Intl. Conf. on Very Large Data Bases, VLDB, pp. 294-303, 1986.

H.Lim, S.Jeong, D.Shin and H.Kim, "The Development of an Automatic Indexing System based on a Thesaurus," Korean Journal of Cognitive Science, Vol.4, No.1, 1993.

G.A.Miller, R.Beckwith, C.Fellbaum, D.Gross and K.Miller, "Introudction to WordNet : An On-Line Lexical Database," in Five Papers on Wordnet," CSL report, Cognitive Science Laboratory Princeton University, 1993.

K.S. Park, "Development of a Thesaurus Management System based on the

Object-Oriented Technique",정보관리학회지, Vol.13, 1996.

J.Sammet and A.Ralstron, "The new computing reviews classification system," Communications of the ACM vol.25, No.1, pp.13-15, 1982.

P.J.Smith, S.J.Shute, D.Glades, and M.H.Cgnell, "Knowledge-Based Search Tactics for an Intelligent intermediary System," ACM Transaction on Information System, pp.246-270, 1987.

S.E.Wright, G.Budin, HandBook of Terminology Management, (http://www.korterm.or.kr)

ETRI, Tachyon User Manual, (http://namoo.etri.re.kr)