지식 추상화와 의미 거리 접근법을 통합한 질의 완화 방법론*

신명근** · 허순영**† · 박성혁** · 이우기***

Relaxing Queries by Combining Knowledge Abstraction and Semantic Distance Approach*

Myung Keun Shin** · Soon Young Huh** · Sung Hyuk Park** · Wookey Lee***

■ Abstract ■

The study on query relaxation which provides approximate answers has received attention. In recent years, some arguments have been made that semantic relationships are useful to present the relationships among data values and calculating the semantic distance between two data values can be used as a quantitative measure to express relative distance. The aim of this article is a hierarchical metricized knowledge abstraction (HiMKA) with an emphasis on combining data abstraction hierarchy and distance measure among data values. We propose the operations and the query ralxation algorithm appropriate to the HiMKA. With various experiments and comparison with other method, we show that the HiMKA is very useful for the quantified approximate query answering and our result is to offer a new methodological framework for query relaxation.

Keyword : Query Relaxation, Approximate Query, Ranking Query Results, Abstraction Hierarchy

논문접수일 : 2006년 02월 20일 논문게재확정일 : 2007년 01월 31일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성 지원 사업(IITA-2005-C1090-0502-0016)의 연구결과로 수행되었음.
** KAIST 테크노경영대학원
*** 인하대학교
† 교신저자

1. 서 론

질의어는 데이터베이스로부터 사용자가 필요로 하는 정보를 얻어내기 위한 도구로 대부분의 데이터베이스 시스템들은 질의 조건과 정확하게 일치하는 데이터에 대해서만 질의 결과를 제공해준다. 그러므로 사용자들은 정확한 질의 조건을 입력하기 위하여 그 데이터베이스의 구조와 내용에 대해서도 충분히 이해하고 있어야만 한다. 심지어 데이터베이스를 잘 숙지하고 있는 사용자들도 만족스러운 결과를 얻을때까지 여러 대안값들을 가지고서 상세한 질의를 반복적으로 시도해야만 한다.

이러한 질의 응답에 대한 한계를 개선하기 위하여, 조건에 정확히 일치하는 결과뿐아니라 보다 넓은 유용한 결과값을 보여줄 수 있는, 이웃 검색[8], 순위종합[9], top-K 질의[4, 5], preference searches [2, 12] 등이 제안되었다. 위의 방법들은 대부분 수치 자료에 대한 조건, 중요도 또는 관련성과 같은 개념을 다룬다. 또한 넓은 범위의 정보 혹은 유사한 응답을 제공할 수 있는 근사값 질의응답이 제안되었다[1, 11, 14]. 근사값 질의 응답의 전형적인 처리 과정은 질의 분석, 질의 완화, 그리고 질의관련 정보 제공의 세 단계로 이루어진다. 질의완화, 질의근사 또는 관련 정보제공을 손쉽게 하기 위해서는 지식 표현 프레임웍이 설정되어야 한다. 여기서 어떠한 유형의 지식 표현 프레임웍이 선택되어야 하는가가 근사값 질의 응답의 성능과 특성에 영향을 미치는 가장 중요한 요인이 된다. 지식 표현기법에 대한 기존의 여러 연구들은 의미적 거리, 추상화 등을 이용하여 수행되어 왔다.

의미적 거리 접근법은 데이터 값들간의 유사도를 표현하기 위하여 의미적 거리 개념을 사용한다[7, 10, 13]. 모든 데이터 값의 순서쌍은 의미적 거리를 갖는다고 가정되므로[13], 이는 간단히 의미적 거리 값에 따라 정렬한 질의결과를 제시해준다. 이는 결과 항목들에 대한 우선순위를 알게 해주므로 유용한 정보를 제공할 수 있다. 하지만 다음과 같은 단점도 있는데, 의미 거리 접근법에서는 다른 범주에 속하는 자료에 대해서 두 데이터 값들 간의 의미 거리가 표로 정리된다. 하지만 모든 순서쌍이 의미 거리를 갖도록 가정되기 때문에, 이러한 방법으로 정리된 표의 크기는 대개의 경우 현실적으로 응용된 분야에 적용되면 엄청나게 커진다. 게다가 표의 크기가 커질수록 유지 비용또한 증가하게 되므로 거리 측정 방법의 견고함을 계속 유지하기 어려워진다.

추상화 접근법(abstraction approach)[6]은 데이터 값들 간의 의미 관계를 표현하는데 효과적인 방법이라고 평가받아온 데이터 추상화 방법(data abstraction method)[3]을 사용한다. 근사값 질의 응답에서 데이터 추상화와 같은 기법은 질의 완화를 위하여 데이터 값들을 연관지어 주는데 유용한 방법이기도 하다. 여러 데이터 값들 중에서 서로 관련있는 데이터 값들끼리 구성해주는 추상화 개념을 구체적으로 실현하기 위하여 타입 추상화 계층(type abstraction hierarchy)[6]에서는 포함의 개념(notions of subsumption), 구성의 개념(notions of composition), 그리고 추상화의 개념(notions of abstraction)을 설명하였고, 다중 레벨 지식 추상화(multi-level knowledge abstraction)를 바탕으로 타입 계층화를 설명하기 위한 완전한 관점을 제시하였다. 그러나 데이터 추상화 방법은 데이터 값들 중에서 명목적 변수에 대해서는 변수간 유사성을 측정하는 척도를 제공하지는 못한다. 그러므로, 추상화 접근법으로는 사용자가 얻어진 질의 결과의 중요성을 판단할 수 없었다. 또한 추상화 접근법이 의미 거리 접근법을 사용하는 시스템에는 적합하지 않았기 때문에, 이 접근법은 질의 결과를 수치화 할 수 있도록 측정 가능하면서 다양한 응용 분야로 확장이 가능한 근사값 질의 응답 시스템을 설계하는데에는 적합하지 못하다.

의미 거리 접근법과 추상화 접근법을 통합하려는 시도도 이루어졌다. 추상화 접근법에서는 수치 도메인에 대한 추상화 기법[6] 및 MTAH(Multiattribute Type Abstraction Hierarchy)[6]를 제안하였다. 그러나, 수치 도메인에 대해 의미 거리를 제시하지 못하였으며, MTAH는 관련있는 도메인들만 모아서 구성할 수 있었다. 즉 의미 거리 접근법과 같이 각각

의 도메인을 독립적으로 처리할 수 없다. Ichikawa는 카테고리화 할 수 있는 자료 처리를 위해 similar graph[10]을 만들고, 모든 링크의 거리를 1로 계산하는 방법을 제안하였다. 그러나, similar grpah은 추상화 계층만큼 구조화 되어 있지 않으며, 모든 링크의 거리를 일률적인 값으로 결정하여 거리 계산의 정밀도를 보장하지 못하였다.

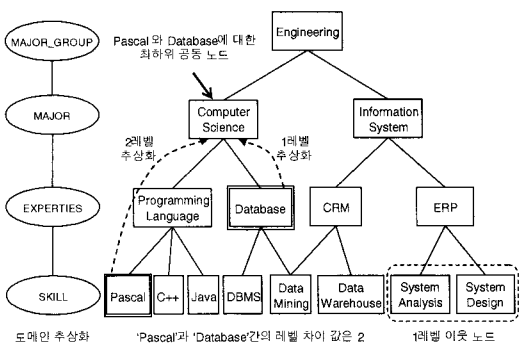
본 논문에서는 추상화 접근법과 의미 거리 접근법을 통합한 새로운 지식 표현 프레임웍을 제안한다. 이것을 계량화된 지식 추상화 계층(hierarchical metricized knowledge abstraction)이라고 부르겠다. HiMKA는 추상화 접근법의 계층화 구조를 사용하여 그 계층 속의 데이터 값들 사이의 계량적인 척도를 제공한다. HiMKA는 표를 이용하여 유사도를 계산하는 척도법보다 유지 보수 비용을 감소시키고, 추상화 값을 사용하여 질의 응답을 제공하는 수준의 풍부한 의미 표현이 가능하도록 한다. 또한 HiMKA는 신뢰성있고 객관적으로 확인된 유사도 척도를 제공하기 때문에 수치 데이터는 물론이고 명목 데이터까지 잘 처리해야하는 시스템에서 사용될 수 있다. 이 논문은 두 데이터 값들 사이의 유사도 거리를 어떻게 계산하는지 보여주고, HiMKA를 이용한 근사값 질의 응답을 어떻게 활용할 것인지 소개한다.

이 논문의 구성은 다음과 같다. 먼저 제 2장에서는 새로운 지식 표현 프레임웍으로써의 HiMKA를 소개하고, 제 3장에서는 질의 완화 과정에 대한 세부적인 사항을 설명하고, 근사값 질의 응답 시스템으로 발전하기 위한 구현 방법에 대하여 알아본다. 제 4장에서는 원형시스템 및 실험 결과를 보여준다. 제 5장에서는 HiMKA의 장점과 관련된 연구와 비교해본 결과에 대해서 알아본다. 마지막 절에서는 이 논문을 요약하고 향후 연구방향을 소개한다.

2. 계량화된 지식 추상화 계층(HiMKA)

2.1 계층에서의 추상화 개념

HiMKA는 데이터 추상화를 이용하여 데이터베이스에 담긴 데이터와 이를 설명해주는 메타-데이터에 대한 다중 레벨 표현을 손쉽게 하는 지식 표현 프레임웍의 한 가지 방법이다. [그림 1]은 Engineering에 대한 추상화 정보를 표현하는 HiMKA의 한 예를 보여주고 있다. 각 계층을 구성하는 값들은 기존 데이터베이스에 존재하였던 자료이거나, 기존에 존재하는 데이터 값들 중에서 의미 관계를 설명하기 위해 추가된 인위적인 값들이다. HiMKA는 두 종류의 추상화 계층인 값 추상화 계층과 도메인 추상화 계층으로 구성된다[1]. 값 추상화 계층 에서는 특정 노드/추상 노드간의 추상화 관계가 존재한다. 레벨에서 임의의 노드는 한 단계 상위 레벨에 위치한 하나의 추상 노드로 일반화 될 수 있다. 이 추상화 관계는 IS-A 관계로 해석될 수 있다. 예를 들어 계층에서 Programming Language가 Computer Science에서 뻗어나온 하나의 가지가 되는 것처럼, Pascal은 Programming Language에서 뻗어나온 하나의 가지가 된다. 이러한 위치 관계중에서 상위 레벨은 그 자체로서 하위 레벨과 보다 더 일반적인 데이터 표현을 제공한다. 그리고 추상화 계층에서 최상위 노드는 가장 추상화된 값을 나타내기도 하고 계층의 대표적인 이름을 의미하기도 한다. [그림 1[에서 최상위 노드는 Engineering이며 전체 추상화 계층의 성격을 설명해주는 역할을 하고 있다.



[그림 1] 레벨 차이값과 이웃 노드

Pascal, C++ 등을 포함하는 리프 노드는 레벨 값 1을 갖는다. 그리고 레벨 값은 하나의 추상화 노드

로 일반화 될 때마다 값이 1씩 증가한다. 특정 노드는 서로 다른 레벨에 위치한 다중의 추상 노드를 가질 수 있다. 예를 들어 Pascal은 레벨 2 위치에 있는 Programming Language와, 레벨 3 위치에 있는 Computer Science를 각각 자신의 추상화 노드로 갖는다. 일반적으로 특정 노드에 대한 n레벨 추상화 노드는 그 특정 노드로부터 n레벨 위에 위치하고 있는 추상 노드이다. 예를 들어 Pascal 로부터 2레벨 위에 위치하고 있는 추상 노드인 Computer Science는 Pascal의 2레벨 추상화 노드가 된다.

임의의 두 노드간의 레벨 차이 n은 두 노드가 공동으로 갖는 상위 추상 노드들 중에서 가장 낮은 레벨에 위치하는 노드와 두 노드 간의 추상화 레벨 값들 중에서 더 큰 값으로 정의된다. n레벨 이웃 노드는 레벨 차이 값이 n인 공동 추상화 노드를 가지는 노드들이다.

2.2 유사도 거리 척도와 계층화에서의 계산 방법

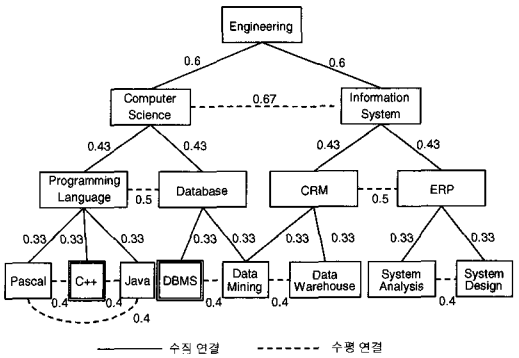
이 부분에서는 유사도 거리 척도의 개념과 최단 경로에 대한 정의를 소개하고자 한다. 그리고 유사도 거리 척도가 필요로 하는 조건을 만족하는 유사도 거리를 정의하려고 한다.

우선 기본거리를 정의해야 한다. *기본거리*는 두 가지 형태의 연결로 구분되는데, *수직 연결*과 *수평 연결*이 있다. 수직 연결은 하나의 특정 노드와 그것의 1레벨 추상화 노드를 이어주고, 수평 연결은 두 개의 서로 다른 1레벨 이웃 노드를 이어준다.

기본 거리는 Wu and Palmer의 척도를 이용하여 정의된다[16]. 이 척도는 그들의 구조적인 관계인 계층안에서 관련된 두 노드가 얼마나 가까운지를 보여준다. z를 x와 y에 대한 최하위 공동 추상화 노드라고 하자. 그러면 x와 y 사이의 기본 거리, $bd(x, y)$는 다음과 같이 정의된다.

$$bd(x, y) = 1 - \frac{2 \times N3}{N1 + N2 + 2 \times N3}.$$

$N1$은 x부터 z까지 경로상의 모든 노드의 갯수이다. $N2$는 y부터 z까지의 경로상에 있는 모든 노드의 갯수이다. $N3$는 z부터 루트까지 경로상의 모든 노드의 개수다. [그림 2]는 기본 거리가 나와있는 HiMKA의 한 예를 보여준다.



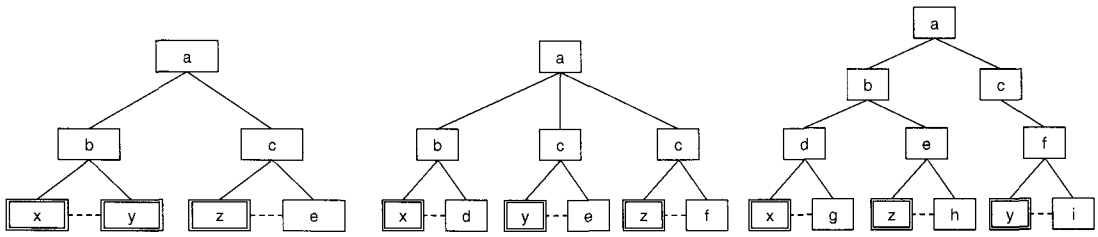
[그림 2] Engineering에 대한 계량화된 지식 추상화 계층(HiMKA)

이제부터 거리 척도에서의 요구 사항을 만족하는 임의의 두 노드 간의 거리를 정의한다. 거리를 계산하는 과정에서 오직 기본 거리만을 고려해보는 것도 가능하지만, 이것 만으로는 임의의 두 노드에 대해서 레벨차이에 비례하는 의미거리를 제공할 수 없다. 왜냐하면 기본 거리를 사용하였을 경우 가끔 레벨 차이 값 3을 가지는 두 노드간에 계산된 거리가 레벨 차이 값 2를 가지는 두 노드간에 계산된 거리보다 더 가깝게 나오기 때문이다. 그러므로 HiMKA에서 임의의 두 노드간의 거리는 최단 경로 상에서 레벨 차이 값과 기본 거리를 사용하여 공식화되어야 한다.

정의 1. (거리) *HiMKA에서 임의의 두 노드 x와 y 간의 거리 $D(x, y)$는 다음과 같이 정의된다.*

$$D(x, y) = x, y\text{의 레벨 차이 값} - 1 + \min_{x, y \in N}(average(\Sigma_{\forall x, y} bd(x, y))).$$

위 식에서 첫 번째 항은 두 노드의 레벨차이에 따라 거리값이 구분되도록 해주며, 두 번째 항 '-1'은 $D(x, y)$의 값이 0부터 시작하도록 보정해 준다. 세



(a) 상황 1 (b) 상황 2 (c) 상황 3

[그림 3] z의 위치에 대한 세 가지 상황

번째 항은 두 노드 x, y를 연결하는 여러 경로 중 기본거리의 합이 가장 작은 경로의 평균값이다. 정의 1의 거리 척도를 사용하면, HiMKA의 임의의 두 노드에 대한 거리는 계산되어 정해질 수 있고, 그 결과들은 레벨 차이 값에 따라 그룹으로 묶인다.

정리 1. *x와 y 보다 더 높은 2 이상의 이웃 노드 값을 갖는 z에 대해서, 정의 1 (거리)에서의 D는 다음을 만족한다.*

$$D(x, y) < D(x, z) + D(z, y).$$

증명 : z의 위치에 대한 세가지 상황에 따라 다음과 같이 나누어 생각하자.

상황 1 : x와 z의 최하위 공동 추상화 노드가 x와 y의 최하위 공동 추상화 노드보다 높다. 그러므로 x와 y간의 레벨 차이 값은 x와 z간의 레벨 차이 값보다 작다. 보조 정리 2에 의하면, $D(x, y) < D(x, z)$이다. 따라서,

$$D(x, y) < D(x, z) + D(z, y).$$

상황 2 : x, y 그리고 z의 최하위 공동 추상화 노드가 x와 y의 최하위 공동 추상화 노드보와 동일하다. n을 x, y, z간의 레벨 차이 값이라고 하면 $n-1 < D(x, y) \leq n$, $n-1 < D(x, z) \leq n$ 그리고 $n-1$ $D(z, y) \leq n$. 따라서 $2 \leq n$에 대해서 다음이 성립한다.

$$D(x, y) < D(x, z) + D(z, y).$$

상황 3 : x와 z의 최하위 공동 추상화 노드가 x와 y 의 최하위 공동 추상화 노드보다 낮다. 이 경우 x와 z 간의 레벨 차이 값은 1 보다 크다. 그리고 x와 y 간의 레벨 차이 값은 z와 y간의 레벨 차이 값과 같다. 보조 정리 2에 의하면 아래와 같다.

$$D(x, y) < D(x, z) + D(z, y).$$

2.3 데이터 모델과 연산 방법

이 절에서는 HiMKA를 관리하기 위한 데이터 모델과 여러 가지 연산에 대해서 소개한다. <표 1>은 지식 데이터 베이스인 두 개의 테이블(DOMAIN_ABSTRACTION, VALUE_ABSTRACTION)에 대한 스키마를 보여주고 있다. 이 지식 데이터 베이스를 통해서 HiMKA를 표현할 수 있다. DOMAIN_ABSTRACTION 테이블은 도메인 추상화 계층의 의미에 대해서 기록한다. 하나의 'domain' 항목은 고유의 이름과, 그것의 'abstract_domain', 'HiMKA' 이름, 그리고 그 HiMKA에서의 위치를 나타내는 추상화 'level'을 갖는다. VALUE_ABSTRACTION

<표 1> HiMKA를 관리하기 위한 지식 데이터베이스 스키마

테이블명	컬럼명	키
DOMAIN_ABSTRACTION	domain abstract_domain HiMKA level	*
VALUE_ABSTRACTION	specific_node domain abstract_node	* *

〈표 2〉 질의 완화를 위한 HiMKA 연산

연산	인수	설 명
DOMAINOF	*r* *a*	테이블 *r* 속성 *a*의 도메인을 반환해준다.
ABSTRACTNODE	*n* *D* *l*	도메인 *D* 상의 노드 *n*에 대한 *l*레벨 추상화 노드를 반환한다. 만약 *n*이 최상위 노드인 경우에는 null 값을 반환한다.
SPECIFICNODE	*n* *D* *l*	도메인 *D* 상의 노드 *n*에 대한 *l*레벨 구체적 노드를 반환한다. 만약 *n*이 리프라면, null 값을 반환한다.
DISTANCE	*n*1 *D*1 *n*2 *D*2	도메인 *D*1상의 노드 *n*1과 도메인 *D*2상의 노드 *n*2간의 거리를 반환한다.

테이블은 HiMKA의 추상화 구조와 수직 거리를 표현한다. 입력받은 'domain'에 대해서, 'abstract_node'는 'specific_node' 의 추상화된 노드이고, 수직 거리는 'distance'이다.

이러한 테이블을 이용하여 <표 2>와 같은 연산을 생각할 수 있으며, 이 연산을 이용한 질의 완화 방법은 다음장에서 설명될 것이다.

3. HiMKA를 이용한 질의 완화

질의 완화는 원래의 질의에다 추가적인 정보를 포함하기 위해 검색 조건을 완화하여 응답의 범위를 넓혀주는 방법을 통해서 가능해진다. HiMKA에서는 추상 노드와 이것에 대한 구체적 노드가 개념적으로 동일하게 여겨지는데, 그 이유는 이 두 노드가 IS_A 관계이기 때문이다. 개념적으로 동일하다는 것은 다음의 세 가지 종류의 개념을 의미한다. 첫째로, 추상 노드는 의미적으로 자신의 구체적인 노드들을 포함한다. 둘째로, 추상 노드는 자신의 구체적인 노드들의 구조적으로 일부분으로 구성된다. 셋째로, 추상 노드는 자신의 구체적인 노드들에 대한 상위 레벨 표현이 된다. 반면에, 이웃 노드들 간에는 각 이웃 노드들이 동일한 추상 노드를 가지고 있기 때문에 근사적으로 동일하게 여겨진다. HiMKA는 체계적인 방식으로 근사적으로 동일하면서 개념적으로도 동일한 응답을 찾는 방법에 사용될 수 있다. 유사한 결과(*similar-to*)를 찾는 연산에 대한 기호로 '=?'를 사용하여 유사 질의를 표현하겠다[2, 10]. 유사 질의는 SQL 구문의 *where* 절에서 '=?'를 사용하여 간단하게 구체화된다. 2 또는 그 이상 레벨의 이웃 노드를 찾는 것을 표현하기 위하여 유사한 결과를 찾는 연산을 확장시킨 '=#?'을 사용하겠다. 여기서 #은 1보다 큰 수치 값이며, 찾고자 하는 이웃의 레벨 값을 가리킨다. 예로, HiMKA에서 '=2?' 표현은 시스템이 2레벨 이상의 이웃 노드를 찾도록 하는 것이다.

〈표 3〉 인사 데이터베이스 스키마

테이블명	컬럼명	키	테이블명	컬럼명	키
EMPLOYEE	Id name dep title	*	COLLEGE_MAJOR	id major entrance_date graduate_date	*
EMPLOYEE_SKILL	Id skill level	*	SKILL_FOR_TASK	task required_skill	* *
			EXPERTISE_FOR_TASK	task required_expertise	* *

이상의 질의 응답 과정을 테스트하는 단계에서 편의를 위하여 단순화시킨 인사 데이터베이스를 사용하였다. 먼저 인사 데이터베이스는 <표 3>과 같이 정의되어 있다. EMPLOYEE 테이블은 직원의 현재 직업에 대한 지위 정보를 제공하고, EMPLOYEE_SKILL 관계는 직원이 가지고 있는 기술을 제공한다. COLLEGE_ MAJOR 관계는 대학 교육 레코드를 포함하고 있고, SKILL_FOR_TASK 관계는 특정 업무에 대해 필수적인 기술을 나타낸다. 마지막으로 EXPERTISE_FOR_TASK 관계는 각각의 업무와 그 업무를 위해 요구되는 전문성간의 관계를 규정한다.

3.1 질의 완화 알고리즘

이번 절에서는 질의 완화 알고리즘을 보여준다. 이 알고리즘은 원본 질의문을 입력 받아 질의 완화를 수행하고 완화된 질의문을 반환한다.

```
Input :  original approximate query    Q
Output : relaxed query    Q'
(1)  translate_query(){
(2)     condition type t ;
(3)     int l ;   // search level
(4)     Q' = Q ;
(5)     for each approximate condition Ci in Q' {
(6)        if Ci is a selection condition {
(7)            (t, l) = alalyze_selection(Ci) ;
(8)            if t == 'approximate query' {
(9)               Ci' = generalize_condition(Ci, l) ;
(10)              Ci'' = specialize_condition(Ci', l) ;
(11)              replace Ci with Ci'' in Q' ;
(12)           }
(13)           if t == 'conceptual query' {
(14)              Ci' = specialize_condition (Ci', l) ;
(15)              replace Ci with Ci' in Q' ;
(16)           }
(17)       }
(18)       else { // Ci is a join condition
(19)            (t, l) = alalyze_join(Ci) ;
(20)            Ci' = two attributes of Ci are
                      appropriately joined with
                      ABSTRACTION ;
(21)            replace Ci with Ci' in Q' ;
(22)       }
(23)    } // for Ci
(24)    return Q' ;
(25)  } // translate_query()
```

이 알고리즘은 원본 질의의 모든 유사 조건에 대한 질의 유형을 분석한다(행번호 6, 7, 19). 유사한 선택 질의를 위해서는 먼저 목표 값이 일반화시킨 다음, 완화된 질의를 얻기 위하여 다시 구체화 된다(행번호 9, 10). 개념적 선택 질의를 위해서는 목표 값을 구체화하는 작업을 통하여 완화된 질의가 얻어진다(행번호 14).

3.2 질의 완화 방법

유사 선택은 목표값 자신뿐만 아니라 그것과 근사적으로 동일한 값도 제공한다. 예를 들어 필수 기술로 Java와 DBMS를 가지고 있는 적절한 고용인 후보자가 채택되지 않거나, 조건을 만족하는 후보자들이 정원을 채울 수 있을 만큼 충분한 수치의 인원보다 부족할 경우, 검색의 범위를 확장시켜 관련된 기술을 가진 다른 사원 후보자들이 얻어지는 것을 필요로 한다. Java와 DBMS 기술을 가진 사원을 검색하는 유사 선택 질의는 다음과 같이 표현될 수 있다.

Q_0 : skill =? 'Java' and skill =? 'DBMS'

유사한 결과를 제공하는 연산자가 의미가 있는 것이 된다면, 그 항목과 구체화된 값 모두 같은 도메인 안에 존재해야 한다. 이러한 관점에서 보면 위 질의에서의 두 도메인이 SKILL로 동일하기 때문에 유사 선택은 유효하다. 원본의 유사 질의는 2.3절에서 소개한 두 연산, ABSTRACTNODE( )와 SPECIFICNODE( )를 사용하여 완화될 수 있다. 첫 번째 단계로, 목표 값인 Java와 DBMS가 ABSTRACTNODE( ) 연산에 의하여 각각 Programming Language와 Database로 일반화된다.

Q_1 : skill is-a ABSTRACTNODE('Java', 'SKILL', 1) and
skill is-a ABSTRACTNODE('DBMS', 'SKILL', 1)

Q_2 : skill is-a 'Programming Language' and
skill is-a 'Database'

다음 단계로, SPECIFICNODE( ) 연산을 사용하여 질의 조건은 {Pascal, C++, Java}와 {DBMS, Data Mining}로 완화되어 질의 조건이 완화된 Q_r이 만들어 진다. 이렇게 완화된 질의는 일반적인 SQL 질의로 응답될 수 있다. 완화된 질의 결과로, 시스템은 Java와 DBMS 기술을 가진 사원뿐만 아니라, 추가적으로 Programming Language와 Database 이내 범위에서 요구되는 기술을 가진 사원도 검색하여 그 결과를 반환해준다.

Q_3 : skill in SPECIFICNODE('Programming Language', 1, 'EXPERTISE') and
skill in SPECIFICNODE('Database', 1, 'EXPERTISE')

Q_4 : skill in SPECIFICNODE {'Pascal', 'C++', 'Java'} and
skill in SPECIFICNODE {'DBMS', 'Data Mining'}

두 번째 예로서, 개념적 결합 조건에서의 두 속성은 서로 다른 도메인을 가질 수 있으며 이로 인해 다른 추상화 레벨 안에 위치할 수 있게 된다. 인사 데이터베이스의 예에서 볼 수 있었던 것처럼 EXPERTISE_FOR_TASK 관계는 업무에서 요구되는 전공을 규정한다. 요구되는 전문성 항목에 대한 도메인은 EXPERTISE이고, 이 도메인은 EMPLOYEE_SKILL 관계에서의 기술 항목에 대한 도메인 보다 더 일반적이다. 이와 같은 방법으로, 사용자는 소프트웨어 디자인 업무와 같은 특정 업무를 수행하기 위해 요구되는 전문성 분야에 속하는 기술을 가진 사람을 찾을 수 있다.

원본 질의 Q_0의 결합 조건에서 x.required_expertise 속성의 도메인은 EXPERTISE이며 s.skill 속성의 도메인은 SKILL로 서로 다른다. 두 속성의 도메인은 다르지만, EXPERTISE라는 하나의 도메인이 SKILL에 대한 추상 도메인이 되기 때문에, 그 질의는 개념적 결합 질의로서 유효하다.

Q_0 : From EMPLOYEE_SKILL s,
EXPERTISE_FOR_TASK x
Where x.task = "Software Design" and
x.required_expertise =? s.skill

질의 완화의 첫 번째 단계로 하위 도메인에 있는 s.skill 속성을 ABSRTACTNODE() 연산을 이용하여 추상화하면 Q_1의 질의문이 된다.

Q_1 : From EMPLOYEE_SKILL s,
EXPERTISE_FOR_TASK x
Where x.task = "Software Design" and
x.required_expertise =?
ABSTRACTNODE(s.skill, 'SKILL', 1)

다음으로 VALUE_ABSTRACTION 테이블이 구체적인 값과 추상 값으로 구성된 쌍을 제공하기 때문에, 동일 추상 노드를 토대로 하여 두 테이블을 연결(join)하는 것은 VALUE_ABSTRACTION테이블을 이용하여 실행될 수 있으며, 최종적으로 Q_r과 같이 완화된 질의문을 얻을 수 있다.

Q_r : From EMPLOYEE_SKILL s,
EXPERTISE_FOR_TASK x,
ABSTRACTION a
Where x.task = "Software Design" and
x.required_expertise = a.abstract_node
and a.specific_node = s.skill

4. 원형 시스템 및 실험

여기서는 HiMKA를 바탕으로 지원자의 기술, 연봉, 그리고 나이를 고려하여 경력직을 찾아주는 시스템 설계를 구현하였다. 이 시스템은 Microsoft SQL Server 7.0을 데이터베이스 서버로 사용하였고, 사용자 인터페이스는 Microsoft Visual C++ 6.0으로 프로그램을 만들었다. 유사 질의들로 구성된 집합의 실행을 통하여 이 원형 시스템의 성능을 측정해보았다. 실험을 위하여 Intel Xeon MP와 2Gb 메인 메모리 환경에서 Microsoft Windows 2000 Server를 사용하였다. 그리고 기술 도메인 상에서 약 200개의 리프 노드를 구성하였고, 인위로 100,000건의 구

인 기록을 만들었다.

〈표 4〉 C++에서 다른 리프 노드들 까지의 거리

이 웃	거리	레벨 차
C++	0.00	1
Java	0.10	1
Pascal	0.40	1
DBMS	1.39	2
Data Mining	1.39	2
Data Warehouse	2.29	3
System Analysis	2.38	3
System Design	2.38	3

대개의 경우 질의 완화 연산의 수행시간으로 0.2초 정도가 걸린다. 데이터베이스 검색은 1.0초 정도가 걸린다. 그리고 노드간 유사도 값을 나타내는 유사도 거리를 계산해주는 경우 0.3초 정도가 요구된다. 따라서 질의 완화와 의미 거리 계산에 필요한 시간 비용은 경미한 수준이다. 유사도 거리 측정에 대한 품질 수준을 평가하기 위하여 모든 리프 노드들 중에서 임의의 두 노드들 간의 유사도 거리를 계산해보았다. 예를들어, <표 4>에서는 [그림 2]에서의 HiMKA에 대해 'C++'에서부터 다른 모든 리프값들까지의 유사도 거리를 보여주고 있다. 이렇게 정리된 유사도 거리들은 레벨 차를 기준으로 묶여진다. 이것은 모든 노드들을 가지고 수행되는 유사도 거리 측정 방법이 일관성있는 결과를 얻을 수 있도록 보장해주어, 시스템이 높은 품질의 유사도 측정법을 갖도록 해준다.

5. 관련 연구와의 비교

의미 거리 접근법을 사용하는 질의 응답 시스템은 질의 결과로 목표 값과 이웃 값들 간의 명목적인 측정을 가능하게 하기 때문에, 사용자들은 유사 값들을 보다 효과적으로 검색할 수 있다. 그러나 카테고리화 되는 데이터들은 모든 데이터 값의 순서쌍들 간의 유사도 거리를 저장하기 위하여 이차원의 표가 작성되어야 하므로 이러한 데이터들에 적용되는 의미 거리 접근법은 다음의 두 가지 문제를 갖는다. 첫째, 목표값의 이웃값을 찾기 위하여 시스템은 목표값과 관계가 있는 모든 레코드들을 검색하여야 한다. 둘째, 새로운 값이 도메인에 추가될 경우 새로이 추가된 값과 기존에 존재하고 있던 다른 항목 값들 간의 유사도 거리가 추가적으로 고려되어야 한다. 이러한 작업은 사람이 직접 생각하여 연산해야 하는 일로써, 수 많은 값들에 대한 유사도 거리 데이터를 할당해주는 과정에서 유사도 거리 값에 대한 일관성을 떨어뜨린다는 단점이 있다.

본 논문에서 제안하고 있는 HiMKA는 이러한 문제점들을 극복할 수 있는 해결책을 제시하고 있다. 첫번째로, 추상화 계층은 목표값을 위한 이웃값을 손쉽게 찾을 수 있도록 편의를 도모한다. 이것을 위해 요구되는 부분은 목표값의 추상 노드를 명확하게 하고 이 추상 노드에 대한 모든 특정한 노드들을 검색하는 것이다. 게다가 검색 조건을 완화시키기 위하여 목표 값에 대한 상위 레벨의 추상 노드를 선택함으로써 더 넓은 범위의 이웃 값들을 얻을 수 있게 한다. 두 번째로, 계층에 담겨있는 거리 정보는 미리 작성한 표에서 임의의 두 노드 사이의 거리 정보를 얻는 방법보다 훨씬 효율적인 방법을 제공한다. 새로운 값을 추가하는 경우 변동 사항이 계층 내부로 국한되기 때문에 이와 같은 변화가 발생하는 경우 거리 정보에 대한 유지 비용은 최소화된다.

또한 HiMKA는 거리 측정법을 제공해주기 때문에 의미 거리 접근법을 기반으로 하는 시스템과 통합될 수 있다. 유형 추상화 계층[6]과 같이 추상화를 기반으로 하는 접근법은 카테고리화 되는 데이터를 다루는데 적합하다. 그러나 이러한 접근법은 수치, 문자, 시간, 날짜 등과 같은 다른 데이터 유형들을 정확하게 처리할 수 없다. 예를 들면, 유형 추상화 계층에서는 계층을 구성하기 위하여 시간을 아침, 점심, 오후, 저녁의 네 주기로 나누어진다. 그러나 대부분의 경우 시간의 차이를 시와 분으로 표현하는 방식을 고려하는 것이 아침, 점심, 오후 등

〈표 5〉 HiMKA와 다른 접근법의 특성 비교

특징 \ 접근법	의미 거리 [7, 10, 13]	추상화 계층 [6]	HiMKA
의미 지식을 표현하는 기법	◦ 데이터 값 간의 의미 거리	◦ 데이터 값 간의 추상화 관계	◦ 데이터 값 간의 의미 관계 ◦ 데이터 값 간의 추상화 관계
양적인 유사도 측정	예	아니오	예
카테고리화 되는 데이터 처리 수준	나쁨	좋음	좋음
확장성	예	아니오	예

의 방법으로 표현하는 것보다 효율적일 수 있다. 이에 반하여, 의미 거리에 기반한 접근법은 다양한 데이터 유형을 다룰 수 있는 여러 계량법을 가지고 있다. 그러나 의미 거리 접근법은 카테고리화되는 데이터를 처리하는데에 어려움이 있기 때문에 단점이 있다.

또한 HiMKA는 사용자에게 잘 구성된 데이터 검색 기능을 제공한다. 사용자는 이 기능을 사용하여 상위 레벨의 추상 구조로 전체 도메인 값을 볼 수 있으며, 드릴 다운 방식으로 세부 정보를 찾아가면서 볼 수 있다. 따라서 사용자가 처음으로 목표 항목을 정할 때, HiMKA는 가장 추상화된 노드를 보여주고나서 사용자가 리프 노드에 도달할때까지 더 세부적인 표본값들을 확인할 수 있게 해준다. 표본 추상 노드를 찾아보고 추상 노드들에 대한 구체적인 노드들을 반복적으로 조사하면서, 사용자는 도메인상의 전체적인 컨텐츠에 대해서 배울 수 있을 뿐만 아니라 검색의 범위를 완화하거나 좁혀 가면서 데이터 검색 절차를 스스로 조절할 수 있다.

HiMKA와 의미 거리, 그리고 추상화 계층을 채택하고 있는 다른 접근법들을 비교한 결과를 다음 표에 정리하였다. 유사 질의 또는 개념적 질의를 다루는 의미 거리 접근법이나 지식 추상화 접근법과 비교했을 때, HiMKA는 응답 결과에 대해 양적인 유사도 척도를 제공하는 것만큼이나 유사 질의, 개념적 질의를 다룰려고 한다. 더군다나 HiMKA는 카테고리화된 자료를 처리하는데 알맞은 방법이고, 의미 거리 접근법을 통합하여 측정 가능하면서 확장 가능한 시스템을 설립하는데도 적합하다.

결론적으로 HiMKA에서는 의미 거리 접근법과 추상화 계층 접근법을 통합하려고 시도하였고, 이러한 노력에 의하여 각 접근법들이 가진 장점을 수용해 두 방법을 상호 보완할 수 있게 되었다.

6. 요약 및 결론

본 논문에서는 데이터베이스에서 검색 질의를 할 때, 근사적 질의 응답을 지원하는 지식 표현 프레임웍의 한 방법으로 HiMKA을 제안하였다. HiMKA는 추상화를 기반으로 설계한 계층 구조로 다중 레벨의 지식을 표현하고, 이웃 노드간의 의미상 유사도를 거리 개념으로 표현해준다. 또한 이웃노드의 거리를 조합하여 임의의 두 노드간의 거리를 계산해주는 거리 측정법을 정의하였다. 이렇게 계산된 유사도 거리는 동일한 레벨 차이 값을 갖는 항목들끼리 그룹지어지고, 유사도에 따라 구분되므로 HiMKA 상의 모든 노드들을 대상으로 적용한 의미 거리 측정법은 일관성있는 계산 결과를 제공해준다. 의미거리 접근법에서는 2차원 테이블에 모든 값들의 거리를 고려하여 계산해주는 반면, HiMKA는 이웃 값들 만을 고려하기 때문에 시스템 유지 비용을 매우 감소시켜준다.

또한 HiMKA는 추상적 질의 응답 관점에서 추상화 수준 정보를 가지고 상호작용 하면서도 유연한 질의 완화를 유도할 수 있으며, 유사도 거리 측정법을 지원하므로 수치 데이터 외에 카테고리화되는 데이터의 계량적 유사도를 다루는데에도 효과적이

다. 그리고 질의 조건으로부터 계산된 유사도 거리에 의하여 유도된 질의 결과에 순위가 매겨지므로 정확한 수치를 바탕으로 응답 결과들을 비교할 수 있게 해준다. 그리고, 목표 레코드와 이웃 레코드의 유사도 거리를 계산할 때, HiMKA가 적용되는 범위는 다른 수치적 분야와 호환 가능하다.

마지막으로, 본 연구에서는 HiMKA를 구현하기 위한 데이터 모형과 연산 방법들을 제안하였다. HiMKA는 검색엔진과, 지식관리시스템, 의사결정지원시스템 등에서 수치적 자료들을 처리하는 것과 같이, 카테고리화되는 자료들을 다루는 경우의 시스템 지식 표현 프레임웍을 위해 적합하다고 볼 수 있다.

향후 연구에서는 실무차원에서의 응용을 위하여 HiMKA를 생성하는 체계적인 절차가 고안되어야 한다. 추가적으로, 근사값 질의응답을 효과적으로 지원하기 위해 최단 이웃 검색(nearest neighbor search), 최상위 k개 자료 검색(top-k selection) 방법들이 제공되어야 한다. 그러나 아직까지도 최단 이웃 질의에 대한 대부분의 연구들이 대부분 수치 자료를 위해서 고안되어왔기 때문에, 카테고리화되는 자료를 다루는 데에는 어려움이 많이 따른다. 앞으로 HiMKA를 사용하여 수치 자료를 처리하는 것만큼 카테고리화되는 자료를 잘 다룰 수 있는 새로운 색인 구조를 설계할 계획을 세우고 있다. 이 구조는 R-tree를 바탕으로 하고 있으며, 각 속성의 범위 정보는 R-tree의 내부 노드에 저장될 것이다. HiMKA의 추상화값은 특정 노드의 범위 정보처럼 사용될 수 있고, 이러한 특성은 다중 차원의 색인 구조를 발전 시키는 방향으로 연구 주제를 발전시킬 것이다. 또한, 새로운 색인 구조를 적용함으로서 최단 이웃 질의를 효과적으로 지원하는 방법도 연구할 예정이다.

참 고 문 헌

[1] 양근우, 허순영, "지식관리시스템을 위한 FAH 기반 전문가 검색 방법론", 「한국경영과학회지」, 제30권, 제1호(2005), pp.129-147.

[2] 이우기, 신광섭, 강석호, "링크내역을 이용한 페이지점수법 알고리즘", 「한국정보과학회논문지 : 데이터베이스」, 제33권, 제7호(2006), pp. 708-714.

[3] Abiteboul, S. and O.M. Duschka, "Complexity of Answering Queries Using Materialized Views," In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Seattle, Washington, USA, (1998), pp.254-263.

[4] Bruno, N., L. Gravano, and S. Chaudhuri, "Top-K Selection Queries over Relational Databases: Mapping Strategies and Performance Evaluation," *ACM Transactions on Database Systems*, Vol.27, No.2(2002), pp. 153-187.

[5] Chakrabarti, K., S. Ortega, S. Mehrotra, and K. Porkaew, "Evaluating refined queries in top-k retrieval systems," *IEEE Transactions on Knowledge and Data Engineering*, Vol.15, No.5(2003), pp.256-270.

[6] Chu, W. and K. Chiang, "Abstraction of High Level Concepts from Numerical Values in Databases," In *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases*, Seattle, USA, (1994), pp.133-144.

[7] Cuzzocrea, A. and U. Matrangolo, "Analytical Synopses for Approximate Query Answering in OLAP Environments," In *Proceedings of the 15rd International Conference on Database and Expert Systems Applications*, Zaragoza, Spain, (2004), pp. 359-370.

[8] Dang, T.K., J. Kung, and R. Wagner, "A General and Efficient Approach for Solving Nearest Neighbor Problem in the Vague Query System," *Lecture Notes in Computer*

Science, Issue.2419(2002), pp.367-378.

[9] Fagin, R., R. Kumar, and D. Sivakumar, "Efficient similarity search and classification via rank aggregation," In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, San Diego, California, USA, (2003), pp.301-312.

[10] Ichikawa, T. and M. Hirakawa, "ARES : A Relational Database with the Capability of Performing Flexible Interpretation of Queries," *IEEE Transaction on Software Engineering*, Vol.12, No.5(1986), pp.624-634.

[11] Kanza, Y. and Y. Sagiv, "Flexible queries over semistructured data," In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems*, Santa Barbara, CA, USA, (2001), pp.40-51.

[12] Klein, M. and B. Konig-Ries, "Combining Query and Preference-an Approach to Fully Automatize Dynamic Service Binding," In *Proceedings of IEEE International Conference on Web Services*, (2004), pp.788-791.

[13] Motro, A., P. Anokhin, and J. Berlin, "Intelligent Methods in Virtual Databases," In *Proceedings of the Fourth International Conference on Flexible Query Answering Systems*, Warsaw, Poland, (2002), pp.580-591.

[14] Muslea, I., "Machine learning for online query relaxation," In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, (2004), pp.246-255.

[15] Vrbsky, S.V. and W.S. Liu, "APPROXIMATE-A Query Processor that Produces Monotonically Improving Approximate Answers," *IEEE Transactions on Knowledge and Data Engineering*, Vol.5, No.6(1993), pp. 1056-1068.

[16] Wu, Z. and M.S. Palmer, "Verb Semantics and Lexical Selection," In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, USA, (1994), pp.133-138.