제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003. 5)

유사도를 이용한 질의 확장과 컴포넌트 검색 방법

정대성, 한정수, 김귀정
천안대학교 정보통신학부, 건양대학교 IT 학부
e-mail:f15cc@dreamwiz.com jshan@cheonan.ac.kr
gjkim@konyang.ac.kr

Query Extension and Component Retrieval Method using similarity

Dae-Sung Jung, Jung-Soo Han, Gui-Jung Kim
Division of Information & Comunnication, Cheon-An Univ.
Division of IT, Kon-Yang University

요 약

본 연구는 유의어 매트릭스를 이용하여 질의의 확장을 통한 컴포넌트 검색 과정을 기술하였다. 컴포넌트 검색은 질의를 입력하면 질의의 확장이 이루어지고 컴포넌트 사이의 신뢰도를 측정하여 검색한다. 신뢰도 계산을 위해서는 질의와 컴포넌트 사이에 유사한가를 나타내는 동치관계, 클래스의 가중치와 동치관계 값을 이용한 포함관계, 그리고 유사도를 계산한다. 끝으로 이들 값을 이용하여 신뢰도를 계산한 후 이 신뢰도 값에 의하여 유사 컴포넌트들을 검색하여 유사도 우선순위로 컴포넌트가 검색된다.

1.서론

본 연구의 검색 시스템은 3 가지 형태의 질의형성 방법을 제공한다. 클래스명 자체를 질의로 사용할 있으며, 클래스 개념 범주(Class Concept Category : CCC)의 트리 구조를 브라우징 하여 원하는 클래스를 찾아 질의로 선택하는 방법, 그리고 질의하고자 하는 클래스의 특정 기능을 자연어로 입력한 후 찾아진 클래스 리스트에서 선택하는 방법이 존재한다. 첫 번째 방법은 시스템의 클래스 구조와 기능을 잘 알고 있는 전문가의 경우 사용할 수 있는 방법이고, 두 번째 방법은 특정 클래스를 질의에 포함시키기보다는 개념적으로 유사한 클래스들 중 가장 적합한 클래스를 비교하여 선택할 수 있는 방법이며, 세 번째는 특정 행위나 기능을 포함하는 클래스를 자연어 형식으로 된 키워드로 쉽게 찾을 수 있는 방법이다.

세 가지 경우 모두 최종적으로는 클래스명을 사용하여 질의를 형성하는데, 사용자로부터 가중치 값을 함께 입력받을 수 있다. 입력된 가중치는 질의에 대한 중요도라 볼 수 있으며, 검색될 컴포넌트가 가지고 있어야 하는 클래스에 대한 만족도라 볼 수 있다. 또한 질의 형성 시 여러 개의 클래스명이 주어지면, 연산 방법에 따라 다양한 검색 결과가 나타날 수 있도록 AND 연산 혹은 OR 연산에 따라 퍼지 불리언 연산을 시행하도록 하였다.

따라서 본 연구의 검색 시스템은 사용자 요구에 부합하는 질의를 형성할 수 있도록 다양한 방법을 제공한다. 질의를 형성하는 클래스의 수나 가중치 값을 변경시킬 수 있으며, 질의 형성 방법을 선택할 수도 있다. 즉, 컴포넌트 검색은 퍼지 불리언 형태로 표현된 사용자 질의를 시소러스를 통해 확장하여 질의로 주어진 질의 확장 방법과 그에 의한 컴포넌트 검색에 중점을 두었다.

2. 퍼지논리와 질의 확장

2.1 퍼지 불리언 질의

본 연구는 퍼지 불리언 형태의 질의를 사용하여 각각의 질의어들에 대해 의미적 중요성을 차등 있게 표현할 수 있도록 하였다. 퍼지 불리언 모델은 사용

자 의도에 따라 질의 관련 정도를 부여할 수 있을 뿐 아니라, 개념들 사이의 관계를 정의한 시소러스와 쉽게 통합될 수 있다는 장점이 있다[1][2][3]. 식(1)은 질의를 형성하는 질의어들에 대한 불리언 연산 식이다. AND와 OR는 불리언 연산자이며, c는 하나의 질의어이고, a는 질의 형성시 입력한 퍼지 질의어의 중요도이다. 포괄적인 의미로 질의를 표현하기 위해서는 퍼지 질의어 중요도인 a를 생략할 수 있는데, 이 경우에는 a=*1.0*으로 간주된다.

$$Q=(AND|OR)[c_i:a_i]_{i=1}^{n}, \quad 0\le a\le 1 \quad (1)$$

퍼지 불리언 질의를 3 가지의 질의로 표준화할 수 있다. 단순질의(mono-query), 분리질의(disjunctive-query), 결합질의(conjunctive-query)가 그것인데, 단순질의는 질의로 하나의 질의만 사용된 경우이며, 두 개 이상의 질의어가 있을 경우 분리질의는 OR의 역할을 하고 결합질의는 AND의 역할을 수행한다. 다음은 3 가지 형태의 질의를 정의한 것이다.

단순질의: $q_\ell=[c_1:a_1]$
분리질의: $q_{\ell(\mathrm{EXP})}=OR[q]_{i=1}^{n}$
결합질의: $Q=AND[q_{\ell(\mathrm{EXP})}]_{\ell=1}^{m}$

본 연구에서는 분리질의를 단순질의에 대한 질의 확장의 결과로 정의하였다. 즉, 하나의 단순질의에 대해서 확장된 질의는 모두 OR로 표현된다. 또한, 분리질의로 표현된 확장된 질의어들은 AND 연산을 함으로써 결합질의로 표현될 수 있다. 따라서 모든 퍼지 불리언 질의는 단순질의의 분리질의에 대한 결합질의로 나타낼 수 있다.

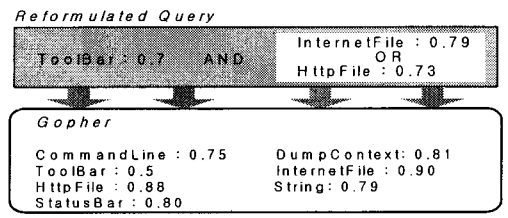
2.2 시소러스에 의한 질의 확장

컴포넌트(i)에 대한 색인 집합(Collection(i))은 클래스명으로 이루어져 있으며, 컴포넌트에 대한 가중치를 가지고 있다. 컴포넌트 i에 대한 색인어 c의 가중치를 $W_{com(i,c)}$= α와 같이 표현하기로 한다. 또한 사용자는 단순질의 q=[c:β]를 입력함으로써 질의어 중요도를 선택할 수 있다. 이때 컴포넌트 i가 질의 q를 만족하는 정도를 γ라 할 때, 이 값은 γ=min(α,β)으로 계산된다. (그림 1의) 'Gopher' 검색을 위한 질의 확장은 'Gopher'의 집합 $\mathrm{Collection}_{(\mathrm{Gopher})}$ = {"CommandLine:0.75,DumpContext:0.81,ToolBar:0.5, InternetFile:0.90,HttpFile:0.88,String:0.79, StatusBar:0.80"}에서 질의 q2=[SocketFile:0.9]의 만족도는 0이기 때문에 초기 질의 "q1=[ToolBar:0.7] AND q2=[SocketFile :0.9]"에 의해 'Gopher' 컴포넌트는 검색될 수 없다.

그러나 퍼지 시소러스에 의한 유의어 테이블에 의해 "q2=[SocketFile: 0.9]"는 <표 1>과 같이 질의 확장될 수 있다. 'SocketFile'은 모든 클래스에 대한 유의값을 가지고 있으며, 이중 유의값이 0.7이상만이 질의 확장의 대상이 된다. 이는 정확도를 적절히 유지하면서도 재현율이 높게 나타나는 임계치를 시뮬레이션 통하여 설정하였다. 따라서 퍼지 시소러스에서 질의 q2='SocketFile'의 질의 집합 $\mathrm{Exp}_{(\mathrm{SocketFile})}$은 {SocketFile:1.0,InternetFile:0.79, HttpFile:0.73}과 같이 구성될 수 있다. 여기서 질의 q2의 'SocketFile' 중요도가 0.9로 설정되면 질의 확장 집합 $\mathrm{Exp}_{(\mathrm{SocketFile})}$의 확장 질의에 대한 유의값이 조절되는데, 질의에 주어진 중요도와 각 확장 질의의 유의값 중 작은 값을 선택한다. 따라서 q2=[SocketFile:0.9]에 대한 최종 질의 확장 집합 $\mathrm{Exp}_{(\mathrm{SocketFile})}$은 {SocketFile: 0.9, InternetFile:0.79, HttpFile:0.73}과 같이 확장된다.

<표 1> 퍼지 시소러스 유의어 테이블

class \ class	...	Internet File	Http File	Archivet	Menu	...
...	...	...	...	...	...	...
SocketFile	...	0.79	0.73	0.65	0.4	...
...	...	...	...	...	...	...



(그림 1) 확장된 질의

(그림 1)은 재형성된 질의를 나타내며, 'Gopher'의 검색 과정은 q1=[ToolBar:0.7] 이고 q2=[InternetFile: 0.79 OR HttpFile:0.73]이므로, 'Gopher'는 q1을 0.5(min(0.5, 0.7)) 정도로 만족하고 있다. 또한 q2에 대해 확장된 각 질의의 중요도와 컴포넌트 가중치에서 적은 값을 선택한 후, 퍼지 OR 연산을 수행한다. 즉, max(min(InternetFile의질의중요도, InternetFile 컴포넌트가중치), min(HttpFile 질의중요도, HttpFile 컴포넌트가중치))로 계산되어, max(min(0.79, 0.90), min(0.73, 0.88))=0.79의 값을 얻을 수 있다. 이는 'Gopher'가 질의 q2를 0.79로 만족하고 있음을 나타낸다. 따라서 'Gopher'가 질의 q1과 q2를 동시에 만족하는 정도는 min(0.5, 0.79)으로 해석되므로, 질의

에 대한 컴포넌트의 만족도는 0.5로 평가되어 최종적으로 'Gopher'는 후보 컴포넌트에 포함되어 검색될 수 있다.

3. 후보 컴포넌트의 유사도 계산

유의어 테이블에 의해 질의가 확장된 후, 이 질의를 만족하는 후보 컴포넌트들이 검색된다. 검색된 컴포넌트의 우선순위를 결정하기 위하여 질의와 컴포넌트들의 유사도를 계산한다[4]. 유사도 계산은 기본적으로 퍼지 함수를 이용하는데, 질의와 대상 컴포넌트를 퍼지 집합으로 인식하고 이 두 집합 사이의 관계값을 퍼지 함수로 계산한다[5][6]. 사용되는 퍼지 함수식을 동치관계식(Equivalence)과 함축관계식(Implication)이다.

과정 1. 질의와 컴포넌트 간의 동치관계 계산
과정 2. 질의와 컴포넌트 간의 함축관계 계산
과정 3. 질의와 컴포넌트 클래스의 만족도 계산
과정 4. 질의와 컴포넌트 간의 유사도 계산

(그림 2)를 기반으로 유사도를 위한 첫 번째 단계는 질의와 컴포넌트를 구성하는 클래스 사이의 유의정도를 유의어 테이블로부터 찾아내는 과정이다. 식(2)은 동치관계를 계산하기 위한 식이다. 이 식은 질의에 나타난 클래스와 컴포넌트에 있는 각 클래스 간의 유의값을 반환한다. 동치관계는 질의와 컴포넌트의 클래스로 구성된 U(질의어 개수)×V(컴포넌트의 클래스 개수)로 이루어진다<표2>. 질의와 컴포넌트 간의 함축관계는 <표3>처럼 질의 중요도와 컴포넌트 가중치 중 큰 값을 반환하여 과정 1의 동치관계와 곱한다. 함축관계 식에서, 질의에 설정한 질의 중요도가 함축관계에 의해 계산된 값보다 작거나 같을 경우에 질의와 컴포넌트의 각 클래스에 대한 교환이 가능함을 의미한다. 함축관계는 식(3)과 같다.

Query
ToolBar : 0.7 AND SocketFile : 0.9
Gopher Component
CommandLine : 0.95, DumpContext: 0.81,
ToolBar : 0.50, InternetFile : 0.90,
HttpFile : 0.88, String: 0.79,
StatusBar : 0.80

(그림 2) 질의와 Gopher 컴포넌트

$$Eq(Query(u), Comp(v)) = SYNON(Query(u), Comp(v)) \quad (2)$$

u : 질의에 있는 질의어 갯수
v : 컴포넌트에 있는 클래스 갯수

<표 2> 질의에 대한 "Gopher"의 동치관계 값

	Command Line	DumpContext	Tool Bar	Internet File	Http File	String	StatusBar
ToolBar	0.769	0.894	1.000	0.654	0.580	0.753	0.912
Socket File	0.752	0.874	0.942	0.901	0.930	0.872	0.890

$$Imp(Query(u), Comp(v)) =$$
$$\max\{w(Query(u)), w(Comp(v))\}[Eq(u,v)] \quad (3)$$

<표 3> 질의에 대한 "Gopher"의 함축관계 값

	Command Line	DumpContext	Tool Bar	Internet File	Http File	String	Status Bar
ToolBar	0.731	0.724	0.700	0.589	0.510	0.595	0.730
Socket File	0.714	0.787	0.848	0.811	0.837	0.785	0.801

만족도(satisfaction value)는 질의와 컴포넌트의 각 클래스가 얼마나 호환성이 있는가를 계산하는 과정이다. 이는 각 컴포넌트 클래스에 대하여 질의어들의 함축관계와 동치관계 값을 곱한 후 그들의 합으로 얻어질 수 있다. 질의와 컴포넌트의 각 클래스 만족집합을 구하면 식(4)과 같다.

$$Sat(Query, Comp(v)) = \frac{[\sum_{u=1}^{U} Imp(u,v) \times Eq(u,v)]}{U} \quad (4)$$

$\sum_{u=1}^{U} Imp(u,v) \times Eq(u,v)$ 는 질의에 대한 한 클래스의 만족정도를 나타내는 값으로써, 하나의 클래스에 대하여 모든 질의어의 함축관계와 동치관계 값을 곱한 후 더해준다. 그러므로 $\sum_{u=1}^{U} Imp(u,v) \times Eq(u,v)$는 함축관계와 동치관계 값이 1일 경우 최대 질의어 수만큼의 값 U를 가질 수 있다. 따라서 이 값을 질의어 수(U)로 나눠줌으로써 0과 1사이의 값을 나타낼 수 있도록 하였다.

$$Sat(Query, Comp(v)) =$$
$$\{0.550, 0.668, 0.725, 0.558, 0.537, 0.567, 0.690\}$$

과정 4에서 만들어진 만족집합에 컴포넌트의 가중치 벡터를 적용하고 모든 요소를 더함으로써 최종적인 질의와 컴포넌트간의 유사도를 계산한다. 가중치는 컴포넌트에 있는 클래스의 모든 가중치 합에 대한 각 클래스 가중치의 비율이며, 가중치 벡터의 요소 개수는 컴포넌트의 클래스 수와 같고 그 합은 1이 된다. 가중치 벡터의 역할은 컴포넌트 내에 있는 클래스들의 매칭 효과를 높이고자 하는데 있다.

다음은 'Gopher' 컴포넌트의 가중치 벡터이다.

$$W=\frac{w(Comp(k))}{\sum_{i=1}^{V} w(Comp(i))}=$$

$$\{\frac{0.95}{5.63},\frac{0.81}{5.63},\frac{0.5}{5.63},\frac{0.9}{5.63},\frac{0.88}{5.63},\frac{0.79}{5.63},\frac{0.8}{5.63}\}$$

'Gopher' 컴포넌트의 가중치 벡터를 사용하여 질의와 컴포넌트의 유사도를 구하면 다음과 같다.

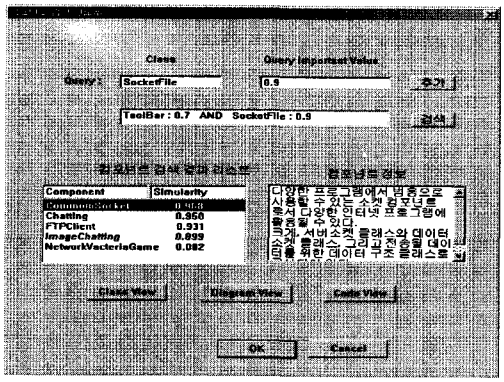
$$Sim(Query,Comp)=$$

$$\sum\{0.550,0.668,0.725,0.558,0.537,0.567,0.690\}\times\{0.169,0.144,0.089,0.160,0.156,0.140,0.142\}$$

$$=0.604$$

$$Sim(Query,Comp)=\sum(Sat\times W)$$

질의에 대해 'Gopher'는 0.604의 유사도를 가지고 있음을 알 수 있다. 검색된 모든 후보 컴포넌트에 대해서 유사도를 계산한 후 가장 높은 값을 가진 컴포넌트 순서로 출력하도록 하였다.



(그림 3) 컴포넌트 검색 윈도우

4. 컴포넌트 검색

컴포넌트 검색은 질의 입력부분, 검색 결과 리스트 출력 부분, 그리고 컴포넌트 정보 부분으로 세분화된다. 질의 입력 부분은 클래스 선택 윈도우로부터 선택된 클래스와 사용자로부터 입력받은 질의어 중요도(query important value)로 구성되며, 1개 이상의 질의어를 선택하길 원할 경우에는 '추가' 버튼을 눌러 다시 클래스 선택 윈도우를 구동시킬 수 있도록 하였다. 본 연구에서는 2개 이상의 질의에 대한 연결은 "AND"로 처리하였는데, 이는 시소러스에 의해 질의가 충분히 확장되었기 때문에 질의 자체를 또다시 "OR"로 처리하면 정확도면에서 효율성이 많이 저하되기 때문에 이를 방지하기 위해서이다.

(그림 3)은 "ToolBar"와 "SocketFile"을 선택하고 중요도를 0.7, 0.9로 주었을 때의 검색 결과이다. 검색 결과 리스트에는 시소러스에 의해 검색된 후보 컴포넌트들이 유사도 순으로 나타나 있고, 선택한 컴포넌트에 대한 정보가 컴포넌트 정보 부분에 제공된다. 효율적인 재사용을 위해서는 검색된 컴포넌트들 중에서 사용자 요구에 가장 적합한 컴포넌트를 선정할 수 있도록 추천하는 기능이 필요하다.

5. 결론

본 연구는 유의어 매트릭스를 이용하여 질의의 확장을 통한 컴포넌트 검색 과정을 기술하였다. 컴포넌트 검색은 먼저 질의를 입력하면 질의에 대한 확장이 이루어지고 그 확장된 질의들로부터 컴포넌트 사이의 신뢰도를 측정하여 검색한다. 신뢰도 계산을 위해서는 질의와 컴포넌트 사이에 유사한가를 나타내는 값인 동치관계(equivalence), 클래스의 가중치와 동치관계 값을 이용한 포함관계(implication), 그리고 유사도(similarity)를 계산한다. 끝으로 이들 값을 이용하여 신뢰도를 계산한 후 이 신뢰도 값에 의하여 유사 컴포넌트들을 검색하여 유사도 우선순위로 컴포넌트가 출력된다.

참고문헌

[1] S. K. M. Wong, V. Raghavan, and P. C. N. Wong, "On Modeling of Information Retrieval Concept in Vector Spaces," ACM Transaction on Database System, Vol. 12, pp. 299-321, 1987.

[2] B. Y. Ricardo and R. N. Berthier, "Modern Information Retrieval," Addison-Wesley, 2000.

[3] H. L. Larsen and R. R. Yager, "The Use of Fuzzy Relational Thesauri for Classification Problem Solving in Information Retrieval and Expert Systems," IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, pp. 31-41, 1993.

[4] M. Moormann Zaremski and J. M. Wing, "Signature matching : A tool using software librarys," ACM Transaction on Software Engineering and Methodology, Vol.4, No.2, pp.146-170, Apr. 1995.

[5] E. Damiani, M. G. Fugini and C. Bellettini, "Aware Approach to Faceted Classification of Object-Oriented Component," ACM Transaction on Software Engineering and Methodology, Vol.8, No.4, pp.425-472. Oct. 1999.

[6] B. Bouchon-Meunier, M. Rifqi and S. Bothorel,"Towards general measures of comparison of objects," Fuzzy Sets System, pp.84, 1996.