클래스의 개념적 분류를 이용한 개념기반 시소러스에 의한 질의 확장

Query Expansion by Concept-based Thesaurus using conceptual classification of Class

김귀정
건양대학교

Kim Gui-Jung
KonYang University

요약

검색 집합에 대한 정확한 지식 없이는 대부분의 사용자가 효율적인 질의 형성에 많은 어려움을 겪고 있다. 이러한 어려움을 극복하기 위한 방법 중의 하나가 초기 질의로부터 더 좋은 질의를 형성해 가는 질의 확장이다. 본 연구에서는 초기 질의의 결과로 검색된 클래스가 가지고 있는 개념을 이용하여 질의를 확장하는 개념 기반 질의 확장 방법을 제안한다.

Abstract

Without detailed exact knowledge of a retrieval collection, most users find it difficult to formulate effective queries. A method to overcome this difficulty is to use query expansion that reformulates better query from initial query. In this paper we propose concept based query evaluation method using concept of class that retrieved from initial query.

Ⅰ. 서 론

소프트웨어 시스템에서 적절한 컴포넌트를 검색하기 위해 효과적인 질의를 형성하는 것은 쉬운 작업이 아니다. 질의하고자 하는 질의문을 완전한 문장으로 작성할 수 없는 상황이라면 적은 수의 키워드로 가장 바람직한 소프트웨어 컴포넌트를 검색할 수 있도록 질의를 변환 또는 확장하는 과정이 필요하다[1]. 이를 위해 시소러스와 같은 지식베이스를 이용한 정보검색 방법들이 많이 제안되었으나, 이처럼 시소러스를 사용하는 일은 많은 검색 시간을 필요로 하며, 심지어는 검색 과정에서 원하는 정보를 찾지 못하는 경우도 발생한다. 본 논문에서는 이 단점을 해결하기 위해 개념을 이용한 질의 확장 방법을 제안하였다. 이 방법은 시소러스를 이용하여 검색에 대한 일정한 정확도를 보장하면서 재현율을 향상시킬 수 있게 한다. 클래스가 가지는 특성에 따라 개념을 설정하고 각 개념과 개념 사이의 관계를 시소러스로 구축하여 사용자 질의와 정확히 일치하는 클래스 뿐 아니라 개념적으로 서로 연관된 유사한 의미를 가지는 클래스까지 검색할 수 있다.

Ⅱ. 관련 연구

객체기반 시소러스[2]는 시소러스에 개념 표현 레벨과 인스턴스 표현 레벨로 구성된 객체지향 패러다임을 적용함으로서 객체들 사이에 존재하는 복잡한 관계성들의 표현에 자동 구축전략을 제공한다. 그러나 이 방법은 효과적인 질의 재형성 과정이 필요하며 실제적으로 응용될 수 있는 검색 시스템의 개발이 필요하다. 적응형 시소러스[3]는 스프레이딩 액티베이션 기반의 유사도 측정 방법을 기반으로 신경망을 이용하여 학습 가능한 시소러스를 제안하였다. 제안한

시소러스는 가중치를 효과적으로 조절할 수 있는 장점이 있지만, 표준화된 형태의 시소러스를 자동으로 추출하는 방법과 보다 적절한 활성화 함수를 개발하는 방법이 요구된다. 계층적 시소러스 시스템[4]은 코드에서 계층적 분류를 위한 범주를 설정하고, 컴포넌트가 행위적 특성에 따라 분류되는 방법을 제안하였다. 컴포넌트 특성은 용어의 쌍으로 구성되며, 소프트웨어 디스크립터(software descriptor : SD)로부터 추출된다. 특성에 따라 계층적으로 분류된 용어의 쌍은 퍼지 시소러스를 이용하여 유의어 사전을 구축하게 된다. 그러나 컴포넌트의 검색이 아닌 멤버함수와 파라미터를 이용한 클래스 검색이기 때문에 클래스가 증가할수록 멤버함수가 기하급수적으로 증가하여 노이즈가 많아진다는 단점이 있다.

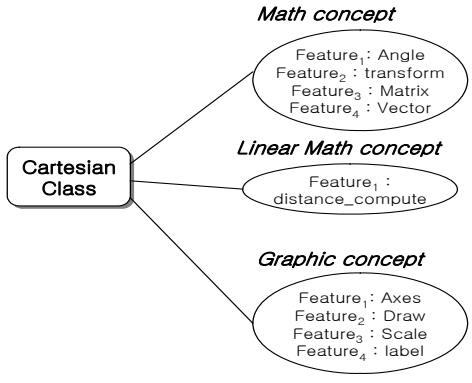
Ⅲ. 개념에 의한 클래스 분류

1. 특성과 개념

본 논문에서 제안한 객체지향 컴포넌트 검색은 개념 기반의 시소러스 질의확장 검색으로써 서로 관련있는 특성들을 일정한 기준에 따라 그룹화하여 여러 개의 개념 그룹으로 구성하는 방식이다. 이를 위해 본 연구에서는 소프트웨어 컴포넌트가 행위적 특성에 따라 분류되는 방법을 제안하였다. 소프트웨어 디스크립터는 컴포넌트에 대한 전체적인 관점을 제공해 주며 이는 컴포넌트의 설명서로부터 만들어진다. 소프트웨어 디스크립터는 컴포넌트 행위를 나타내는 용어의 리스트로 구성된다. 이때, 용어를 '특성(feature)'이라 한다. 또한 패싯 분류를 이용하여 각 특성을 '개념(concepts)'으로 구성하였다. 전통적인 정보 검색 시스템에서 사용된 용어의 통계적 방법에 의한 시소러스 구축은 객체 지향 컴포넌트에 그대로 적용될 수 없기 때문에 특성에 대한 개념을 중심으로 패싯 분류하였다. 즉, 특성은 각 컴포넌트를 나타내는 용어이며, 이 특성은 일정한 기준에 의해 개념으로 분류되어 그룹화되어 진다. 그러므로 하나의 컴포넌트는 여러 개의 특성을 가질 수 있으며, 그 특성이 포함된 개념에 따라서 여러 범주의 개념에 속할 수 있다.

2. 클래스의 개념적 분류

본 연구는 개념을 기반으로 한 시소러스 질의 확장을 제안한다. 객체지향 컴포넌트에 대한 시소러스는 전통적인 정보 검색 시스템에 이용되는 시소러스 구조 및 관계에 있어서 해석의 차이가 있다. 즉, 객체지향 컴포넌트의 특성에 맞는 클래스의 개념적 분류 방법이 필요하다. 이에 따라 본 연구에서는 클래스를 개념에 따라 분류하였다. 클래스가 사용될 수 있는 여러 경험적 상황을 패싯 항목으로 설정하는 패싯 분류방법을 사용하였다. 먼저 각 클래스를 표현하는 특성을 소프트웨어 디스크립터로부터 설정한 후, 모든 클래스로부터 나온 특성을 그룹화하여 개념을 생성한다. 그러므로 하나의 클래스는 그 특성에 따라 하나 이상의 개념을 가질 수 있으며, 이는 검색 시 개념을 이용한 시소러스 질의 확장에 이용된다. 그림 1은 'Cartesian' 클래스가 가진 특성과 개념을 나타낸다. 'Cartesian'은 3개의 개념에 속해있음을 알 수 있다.



▶▶ 그림 1. 'Cartesian' 클래스의 특성과 개념

클래스는 기능과 개념에 따라 여러 개의 범주로 나눌 수 있다. 이 과정은 도메인 전문가에 의해 행해지며 시스템의 응용에 따라 모든 클래스를 최적으로 표

현할 수 있는 개념을 선정한다. 각 개념은 패싯 항목으로 표현되고 이에 따라 클래스가 분류되어지며 시소러스 구축 시 이용되게 된다.

3. 클래스 정보 표현

클래스 정보는 특성, 개념, 도메인 정보, 시소러스 정보로 구성된다. 이를 위해서 클래스들을 구문 분석하여 클래스내의 모든 정보를 상호 관련정보로 구축한다. 또한 각 클래스들의 관계정보도 추출한다. 그림 2는 클래스 정보 표현 구조를 나타낸다. 정보 표현에서 특성과 개념이 리스트로 표현되는 이유는 한 클래스가 여러 개의 특성을 가지고 있고 특성에 대응하는 여러 개의 개념을 가질 수 있음을 의미한다. 이와 같이 표현된 클래스 정보는 소스코드 링크정보와 함께 정보저장소에 저장된다. 클래스의 정보는 스트링(예, CLASS), 클래스명, 개념 리스트, 특성 리스트 순으로 저장한다. 또한 전체 클래스 코드를 나타내기 위한 클래스 단위의 코드 정보와 클래스 질의 표현 데이타베이스에도 링크되도록 하였다. 이는 클래스 수만큼의 반복적 추출과정을 실행하여 정보를 저장한다.

Class-ID : [class name]
- Use : [클래스 질의 표현어 리스트]
- Features : [특성 리스트]
- Concepts : [개념 리스트]
- Domain : [응용 도메인]
- Thesaurus info : [시소러스 정보]

▶▶ 그림 2. 클래스 정보 표현

Ⅳ. 개념적 질의확장

1. 개념 기반 시소러스 구축

시소러스 구축은 개념과 특성들 간의 관련값에 대한 통계적 분석에 의해 이루어진다. 먼저 특성과 개념간의 관련값(FCV)이 얻어진 후에, 이를 이용하여 개념과 개념의 관련값(CCV)을 계산한다. 다음은 시소러스 구축과정이다.

① 개념과 특성으로 이루어진 메트릭스를 구성하여 특성-개념관계값 (Feature-Conceptt Value : FCV)을 계산한다. 특성-개념관계값에 대한 수식은 식(1)에 나타나 있다. 이는 각 개념에 속한 특성의 수는 컴포넌트와 개념과의 관련성을 암시해 준다는 의미에 근거한다.

$$FCV_{l,j} = p_j(l)\frac{feature_{l,j}}{feature_l} \quad \text{식(1)}$$

$FCV_{l,j}$: *Concept j*와 *feature l*과의 관계값
$p_j(l)$: *Concept j*에서의 *l* 번째 *feature*의 발생백분율
$feature_{l,j}$: *Concept j*에서 *l* 번째 *feature*의 발생횟수
$feature_l$: 모든*Concept*에서 *l* 번째 *feature*의전체발생횟수

② FCV matrix를 이용하여 개념과 개념 관련값 (Concept-Concept Value : CCV)을 계산한다. 이는 각 특성에 대한 개념의 매칭정도를 나타내며 계산식은 (2),(3)과 같다.

[표 1] FCV matrix

Feature \ Concept	C_1	C_1	...	C_C
A	a_1	a_2	...	a_c
B	b_1	b_2	...	b_c
...	...	...	...	...
N	N_2	N_2	...	N_c

ⓐ 개념 C1과 C2 사이의 매칭정도를 알아보기 위해 먼저 a1과 a2 사이의 매칭정도 계산

$$ma_{12} = \frac{1}{1+|a1-a2|}, \quad (a1 \neq 0,\ a2 \neq 0)$$
$$ma_{12} = 0, \quad (a1 = 0,\ OR\ a2 = 0) \text{ 또는 } (a1 = a2 = 0) \quad \text{식(2)}$$

C1과 C2의 모든 특성에 대해서 계산

$$M_{12} = \sum_{j=a}^{N} mj_{12} \quad \text{식(3)}$$

ⓑ 모든 개념에 대해서 시행한다.

위와 같은 과정에 의해 최종적으로 개념과 개념 사이의 관련값, 즉 개념간 유의어 테이블이 구성되어 개념을 기본으로 한 시소러스가 구축된다.

2. 질의 확장

시소러스에 의한 개념 확장 과정은 다음과 같다.

① 기본 개념 추출

질의에 해당하는 클래스를 검색한 후 그 클래스의 특성이 포함된 개념을 추출한다. 예를 들어 질의가 'URL'과 'BufferedReader'일 때 이를 만족하는 클래스 'URLReader' 클래스를 검색한다. 'URLReader' 클래스의 특성을 추출한다.

'Exception', 'URL', 'BufferedReader' 그리고 'inputstream' 의 특성이 추출되었고, 이에 대응하는 개념은 'Exception'은 'Debugging' 개념에 포함되고 'URL'은 'Network' 개념에 포함되며'BufferedReader'와'inputstream'은 'BufferedReader' 개념에 포함된다. 그러므로 질의 'URL'과 'BufferedReader'에 의해 선택된 기본 개념은 'Debugging', 'Network', 'BufferedReader'이다.

② 개념 확장

추출된 개념을 시소러스에 의해 확장한다. 기본 개념으로 선택된 'Debugging', 'Network', 'BufferedReader'는 시소러스에 의한 CCV 테이블을 이용하여 개념이 확장된다. 각 개념은 모든 개념에 대한 관련값을 가지고 있으며, 이중 0.7 이상에 해당하는 개념들만이 확장의 대상이 된다. 이는 정확도를 적절히 유지하면서도 재현율이 높게 나타나는 임계치 범위를 시뮬레이션 통하여 설정한 것이다[5]. 표 2는 'Network'에 대한 유의어 테이블의 일부이다. 이중 관련값이 0.7 이상인 'Internet'과 'HttpFile'이 'Network'에 대한 확장 개념으로 선택되어 진다.

[표 2] 시소러스 유의어 테이블

concept / concept	…	Internet	HttpFile	Archive	Menu	…
…	…	…	…	…	…	…
Network	…	0.78	0.72	0.48	0.61	…
…	…	…	…	…	…	…

V. 성능평가 및 결론

본 연구에서는 시스템의 효율성을 평가하기 위하여 검색 효율성의 기준이 되는 재현율과 정확도를 측정하였다. 질의는 임의로 30개를 선정하였고, 시소러스를 사용하지 않은 것과 본 연구에서 제안한 방법으로 시소러스를 사용한 경우의 재현율을 0.1 단위로 변화시키면서 정확도의 변화를 측정한 후, 정확도의 평균을 비교하였다. 표 3은 두 경우의 정확도와 재현율의 비율을 나타낸 것이다. 본 연구에서 제안한 시소러스에 의한 개념 기반 검색은 시소러스를 사용하지 않은 검색에 비해 효율성에 있어서 22.3%(((0.769-0.629)/0.629)*100) 정도 크게 향상되었음을 보여주고 있다.

본 논문은 특정 클래스로부터 개념적으로 서로 연관있는 클래스를 검색하기 위하여 개념 기반 시소러스를 통한 질의 확장 방법을 제안하였다. 제안한 방법은 클래스가 가지는 특성에 따라 클래스를 개념으로 분류하고 모든 개념에 대해 개념별 관련값을 이용하여 개념 기반 시소러스 유의어 테이블을 구성하였다. 초기 질의에 의해 검색된 클래스의 개념은 시소러스를 통해 확장된다. 이로 인해 개념 기반 질의 확장의 효율성이 시소러스를 사용하지 않았을 때의 검색에 비해 효율성에 있어 22.3% 정도 크게 향상되었음을 보여주었다. 본 연구는 개념을 이용한 질의 확장을 통하여 후보 컴포넌트까지 검색하여 컴포넌트 선택의 범위를 넓힐 수 있었으며, 클래스 라이브러리에 대한 개념을 이용하여 컴포넌트들을 검색하고 우선순위로 표현하기 때문에 컴포넌트 조립시 보다 효율적이다.

[표 3] 재현율과 정확도의 비율

Recall	No-시소러스 Precision	개념 기반 시소러스 Precision
0.1	0.68	1.00
0.2	0.70	0.95
0.3	0.78	0.92
0.4	0.75	0.82
0.5	0.72	0.86
0.6	0.63	0.72
0.7	0.59	0.71
0.8	0.54	0.65
0.9	0.48	0.57
1.0	0.42	0.49
평균 Precision	0.629	0.769
향상된 평균 비율	-	22.3%

▌참고문헌▐

[1] F.A.Grootjen, and Th.P. van der Weide, "Information retrieval as a semantics transformation mechanism. a formal theory for latent semantics," Technical Report NIII-R0303, University of Nijmegen, 2003.

[2] 최재훈, 한종진, 박종진, 양재동, "구조적인 시소러 스 구축을 지원하는 객체 기반 정보 검색 모델," 한국정보과학회논문지, Vol.24, No.11, pp.1244-1256, Nov., 1997.

[3] 최종필, 최명복, 김민구, "신경망을 이용한 적응 형 시소러스," 한국정보과학회논문지, Vol.27, No.12, pp. 1211-1218, Dec., 2000.

[4] 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.

[5] 김귀정, 한정수, 송영재, "컴포넌트 검색을 지원하는 퍼지 기반 시소러스 구축", 한국정보처리학회논문지제 10-D권, 제5호, pp.753-762, 2003, 8.