제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

컴포넌트 검색을 위한 효율적 시소러스 구축에 관한 연구

한정수*
천안대학교 정보통신학부
e-mail:jshan@cheonan.ac.kr

A Study on Efficient Thesaurus Construction for Component Retrieval

Jung-Soo Han*
*Division of Information and Communication, Cheonan University

요 약

본 연구는 컴포넌트의 효율적인 검색을 위하여 각 컴포넌트의 코드 정보를 분석하고, 카이제곱 검정 방법을 통하여 분석한 컴포넌트 정보인 term들을 최적화하였다. term의 최적화를 위한 시뮬레이션에서 전체 term 중 약 20%가 제거되었다. 또한 최적화된 term들을 이용하여 term-component 관계를 이용한 매칭, 비매칭 관계 계산을 통하여 term-term 사이의 유의어 매트릭스를 구성함으로써 시소러스를 구축하였다. 시소러스를 통한 검색 결과 최적화 이전의 검색결과와 성능이 비슷하게 나타나 본 연구의 시소러스 구축이 더 효율적임을 알 수 있었다.

1. 서론

최근 소프트웨어 컴포넌트 기반 개발 방법(CBD)과 분류, 검색 방법 등이 많은 발전을 하고 있다. 특히 컴포넌트들을 조립하여 소프트웨어를 개발하기 위해서는 이전의 개발된 컴포넌트들은 관리를 위한 보다 정확하고 많은 정보를 갖고 저장소에 관리되어야 한다. 따라서 새로운 응용 개발에 필요한 재사용 컴포넌트의 조립을 위해서도 많은 검색방법론이 제시되고 있다.

기존의 컴포넌트 재사용을 위한 검색 방법 중 시소러스 구축 연구에서는 컴포넌트에 존재하는 모든 term을 이용하여 시소러스를 구축하였기 때문에 많은 계산 시간과 검색 결과 노이즈가 많이 발생함을 알 수 있었다. 따라서 본 연구는 개발된 컴포넌트들을 효율적으로 검색하기 위하여 컴포넌트 검색을 위한 term의 최적화를 통한 시소러스를 구축하였다. 시소러스 구축을 위한 term의 추출을 위해서는 먼저 SD(Software Descriptor)를 이용하여 컴포넌트를 분석하고 분석된 컴포넌트의 term들을 카이제곱 검정방법을 통하여 최적화하였다. term의 최적화를 위하여 카이제곱 통계량 계산 이전에 공통적이고 노이즈(noise)가 될 가능성이 있는 term들을 제거하여 보다 최적의 term 정보를 갖고 term과 컴포넌트 사이의 관계를 이용하여 시소러스를 구축하였다. 그리고 시소러스 구축을 위해서는 먼저 각 term의 컴포넌트 상에서의 발생 빈도수를 이용한 가중치(weight value)를 구하여 그 관계를 정의 하였다.

따라서 본 연구는 컴포넌트의 검색을 위하여 기존의 시소러스 방법을 개선시켜 term-component 사이의 관계성을 이용한 효율적인 시소러스 구축에 그 목적을 두었다.

2. 관련연구

기존의 시소러스 구축 관련 연구에는 컴포넌트와 term 사이의 관계를 이용한 구축방법[1], 객체지향 관계를 이용한 시소러스 구축방법[4] 등이 있고, 검색을 위한 방법에는 spreading activation[2] 방법과 필터링과 클러스터링(filtering and clustering)[3]을 통한 분류 검색 방법 등이 있다.

컴포넌트와 term 사이의 관계를 이용한 구축 방법은 컴포넌트에 포함되어 있는 모든 term을 이용하여 각 term들을 개념적인 컨텍스트(context)로 분

류하여 시소러스를 구축하였다. 그러나 이 방법은 각 컴포넌트에 term의 수가 너무 많아 컴포넌트의 수가 증가할수록 무한정 증가하는 term의 정보를 이용한 시소러스 구축에는 한계가 있다. 또한 검색 시 다중 컴포넌트에 속한 term이 많기 때문에 노이즈(nose)가 많이 발생한다. 객체지향 관계를 이용한 시소러스 구축은 클래스들을 개념 표현 레벨과 인스턴스 표현 레벨을 이용하기 때문에 많은 컴포넌트들에 대한 개념표현 레벨을 정의하기가 어려운 단점이 있다. spreading activation방법은 term-document사이의 관계를 이용한 유사 컴포넌트 검색방법으로서 document의 증가에 따른 계산이 많아 검색에 걸리는 시간이 늘어나는 단점이 있다. 또한 필터링과 클러스터링(*filtering and clustering*)분류 방법은 컴포넌트들의 클래스들의 구조를 이용한 분류방법이다.

따라서 본 연구는 term-component 사이의 관계를 이용한 시소러스 구축 방법에서 term의 최적화를 위해 카이제곱 검정방법을 사용하여 컴포넌트와 term의 관계를 이용한 시소러스를 구축하였다.

3. 시소러스 구축

3.1 컴포넌트 정보 추출

컴포넌트 정보 추출기(Component Descriptor: CD)는 컴포넌트의 원시코드에 대한 구문분석을 통하여 각 컴포넌트에 포함되어 있는 멤버 함수에 대한 term(method-parameter)들을 컴포넌트의 명과 함께 추출한다. 여기서 사용자 정의 멤버함수와 그 파라미터는 컴포넌트 명과 term-component의 관계가 된다. 이 term-component 매트릭스는 카이제곱 통계량 계산에 이용되어 최적의 term의 수를 구하는데 사용된다. 또한 시소러스 구축을 위해 term과 컴포넌트의 매칭 또는 비매칭 계산에 활용되어 유의어(synonymy) 매트릭스를 구성함으로서 시소러스가 구축되는 정보이다.

3.2 카이제곱 검정

본 연구는 SD로 추출한 컴포넌트 정보를 최적화하기 위하여 카이제곱 통계량을 이용하였다. 최적화의 목적은 예를 들어 추출된 term a가 모든 컴포넌트에 포함되어 있으면 이 term을 이용한 검색은 그 의미가 없기 때문에 이와 같은 term들을 제거하기 위함이다. 식(1)은 카이제곱 통계량으로서 하나의 term에 대한 가중치를 계산한다. 그 결과 각각의 term들은 모두 자신의 가중치(*weight value*)를 갖게 되고, 시뮬레이션을 통한 임계치를 설정하여 term들을 제거한다. 시뮬레이션의 결과 전체 term중에 약 20% 정도가 제거되었다.

$$x^2(t,c)=\frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)}$$

t : term
c : component
A: t와 c가 동시에 발생한 횟수
B: t는 발생했지만 c는 발생하지 않은 횟수
C: t는 발생하지 않고 c만 발생한 횟수
D: t와 c 모두 발생하지 않은 횟수
N: 전체 컴포넌트의 수

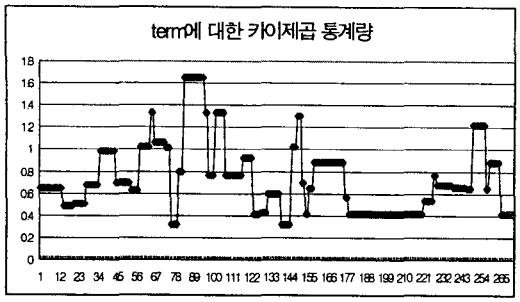
3.3 시뮬레이션

시뮬레이션을 위한 컴포넌트 정보 분석 자료는 <표1>에서처럼 23개의 컴포넌트를 가지고 분석하였다. 각 컴포넌트는 term들을 포함하고 있고 이들 term들은 서로 중복된 term들도 존재한다. (그림 1)은 각 term들에 대한 식(1)에 의한 카이제곱 통계량의 결과에 대한 전체 term의 분포도를 나타낸 것이다. 처음 초기화된 통계량의 분포도는 (그림 1)과 같다. 각 term의 가중치를 카이제곱 통계량을 통하여 분석해 보면 단측 검정에서 유의수준에 대한 제거기준이 (그림 2)에서와 같이 3가지의 부류로 나타난다. 본 연구에서는 중간 정도인 19.85% 제거 량을 기준으로 채택하였다. 그 결과를 분석해 보면 각각의 term들이 컴포넌트에 포함되면서 컴포넌트를 식별하는 동시에 컴포넌트를 대표할 수 있는 최적의 term들로 구성된다.

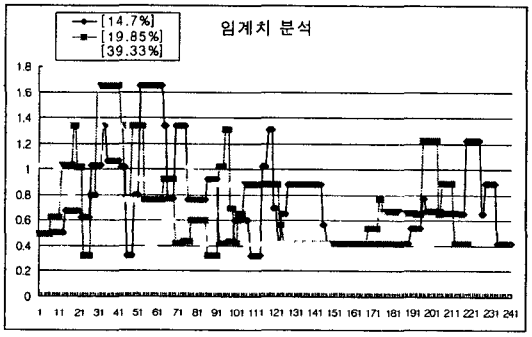
제거된 term들은 주로 많은 컴포넌트에 존재하는 term들로서 이들 term을 이용한 검색은 유사하지만 정확하지 않는 즉, 노이즈가 되는 컴포넌트들이 검색될 수 있는 위험을 포함하고 있다. 따라서 이러한 term들을 제거함으로서 보다 정확한 컴포넌트들을 검색할 수 있도록 도와준다.

<표1> 컴포넌트 분석 자료

	초기값	제거한 후의 값
전체 컴포넌트의 수	23	23
전체 term의 수	272	219
중복된 term의 수	41	18
중복된 term의 최대 수	17	14



(그림 1) 카이제곱 통계량



(그림2) 임계치 분석

3.4 시소러스 구축

검색 시스템에서 자동적인 시소러스 구축은 컴포넌트의 term에 대한 통계적 분석으로 이루어진다. term-component의 초기화 매트릭스는 식(1)에서 얻은 가중치로 이루어지고, <표2>와 같은 매트릭스가 구성된다. 여기서 a1, a2, ..., nc는 가중치를 나타내고, m_{1ab}, ...m_{cab}는 매칭 비매칭 값을 의미한다. 그리고 이들 가중치를 이용하여 시소러스 구축을 위한 term과 component 사이에 대한 각각의 매칭, 비매칭 계산이 이루어지고 이들의 값을 이용하여 term-term, 사이의 유의어 매트릭스를 구성한다.

<표2> Term-Component Matrix

Term \ Component	C1	C2	...	Cc
A	a1	a2		ac
B	b1	b2		bc
...				
N	n1	n2		nc

시소러스 구축 과정은 다음과 같다.

1) term-component matrix 구성($N_{term} \times C_{component}$)
2) $1 \le a \le N_{term}$ 와 $1 \le C_i \le C_{component}$ 사이에서 식(1)을 이용하여 각각의 가중치 계산(a1,a2.....,nc)
3) 각 term을 기준으로 매칭, 비매칭 계산
   ① term사이의 매칭 계산

$$m_j = \frac{1}{1+\Delta_j} \quad , \quad \Delta_j = |a_j - b_j|$$

   ② term사이의 비매칭 계산

$$m_j^* = \frac{\Delta_j}{1+\Delta_j} \quad , \Delta_j = |a_j - b_j|$$

①과 ②에서 각각의 합을 구하면 다음과 같다.

$$M = \sum_{j=1}^{c} m_j$$

$$M^* = \sum_{j=1}^{c} m_j^*$$

4) 3)의 내용으로부터 각 컴포넌트를 기준으로 컴포넌트 사이에 대한 매칭, 비매칭 계산을 유도할 수 있다. 즉, C1과 C2 사이의 매칭(a1과 a2 값 이용) 계산을 이용하여 m_j'를 구할 수 있다. mj*'도 마찬가지이다. 즉,

$$m_j' = \frac{1}{1+\Delta_j} \quad , \quad \Delta_j = |a_j - a_{j+1}|$$

$$m_j^{*\prime} = \frac{\Delta}{1+\Delta_j} \quad , \quad \Delta_j = |a_j - a_{j+1}|$$

또한, 각각의 합인 M'와 M*'도 구할 수 있다.

5) 유의어(synonymy) 매트릭스

각 계산식에 의하여 유의어 값 f는 식(2)와같이 계산된다. 각각의 f 값은 결과적으로 term과 term 사이의 유의어 값이 되어 <표3>과 같은 유의어 매트릭스 즉 시소러스가 구축된다.

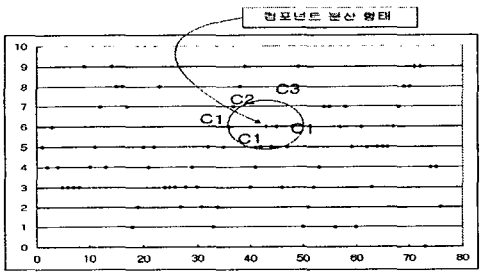
$$f = \min\left(1, \frac{1}{2}\left|\left(\frac{M-M^*}{M+M^*} + \frac{M'-M^{*\prime}}{M'+M^{*\prime}}\right)\right|\right) \qquad (2)$$

<표3> 유의어 매트릭스

term \ term	a	b	...	n
a	f_{a1}	f_{a2}		f_n
b	f_{b1}	f_{b2}	...	..
...				..
n	...		...	f_{nn}

4. 컴포넌트 검색

term-term 사이의 유의어 매트릭스를 이용한 컴포넌트 검색은 먼저 쿼리(term)를 입력하면 쿼리와 컴포넌트 사이의 신뢰도를 측정하여 검색한다. 신뢰도 계산을 위해서는 쿼리와 컴포넌트 사이에 얼마나 유사한가를 나타내는 값인 동치관계(equivalence), term의 가중치와 동치관계 값을 이용한 포함관계(implication), 그리고 유사도(similarity)를 계산한다. 끝으로 이들 값을 이용하여 신뢰도를 계산한 후 이 신뢰도 값에 의하여 유사 컴포넌트들을 검색한다[1]. (그림 3)에서처럼 컴포넌트가 각각의 term에 대하여 분산되어 있고, term의 신뢰도를 이용하여 신뢰도 값을 중심으로 일정한 범위 내의 컴포넌트들을 검색하고, 범위내의 컴포넌트의 빈도에 따라 우선 순위로 컴포넌트가 검색된다.



(그림 3) 컴포넌트 분산

<표 4> 기존의 시소러스와 비교

기능 / 방법	기본구성 단위	시소러스 방법	컴포넌트 검색
No 시소러스	term	string matching	직접연결검색
객체지향 시소러스[4]	term	개념 정의와 객체관계	관계에 의한 유사검색
계층적 시소러스[1]	term	term의 유의어사전	context에 의한 검색
제안한 시소러스	term	최적의 term을 이용한 통계적 유의어 사전	신뢰도를 이용한 검색

5. 평가

본 연구는 컴포넌트의 검색을 위한 효율적 시소러스 구축에 목적을 두었다. 먼저 컴포넌트의 특징(term)을 추출하여 term의 수를 카이제곱 통계량을 이용하여 최적화하고, 이 최적화된 정보를 이용하여 term-component 사이의 가중치를 이용한 매칭 정도와 비매칭 정도를 계산함으로서 term-term 사이의 유의어 매트릭스를 구성하였다. <표4>에서 제안한 방법은 기존의 방법[1] 보다 term의 수를 최적화함으로서 노이즈(noise)를 줄일 수 있었으며, 시소러스 구축이도 효율성이 높음을 알 수 있었다. 또한, 객체지향 시소러스[4]에서 사용하는 객체관계는 연관, 집단, 일반, 분류 관계를 이용하기 때문에 유사하지만 관계가 먼 컴포넌트가 검색될 가능성이 높다.

6. 결론

본 연구는 컴포넌트의 특성 정보를 분석하여 추출한 term의 수를 카이제곱 통계량을 이용하여 최적화함으로서 보다 효율적인 term들을 구성하여 시소러스를 구축하였다. 최적화는 약20%정도 이루어졌다. 또한 시소러스 구축을 위해 컴포넌트와 term 사이의 매칭, 비매칭 계산을 활용하여 각 컴포넌트에 term이 어느 정도의 중요도를 나타내는지에 대한 중요도를 계산한 후 유의어 매트릭스를 구성하여 시소러스를 구축하였다. term에 의한 검색은 term과의 관련성 계산으로 신뢰도를 측정하여 일정 범위 내의 컴포넌트 빈도에 따라 우선순위로 검색된다. 기존의 시소러스와 비교해본 결과 최적화된 term을 이용하였기 때문에 노이즈가 적게 나타나고, 성능은 비슷한 결과를 낳았다.

앞으로의 연구방향은 term들을 자연어처리를 통한 검색방법으로 발전시키고, 검색된 컴포넌트의 효율적 조립 방법에 관하여 연구가 되어야할 것이다.

참고문헌

[1] E. Damiani, M. G. Fugini, and C. Bellettini, "A Hierarchy-Aware Approach to Faceted Classification of Object-Oriented Components ", ACM Transaction on Software Engineering and Methodology, Vol. 8, No. 4, October 1999, PP. 425-472.

[2] Scott Henninger, "Information Access Tools for Software Reuse", System Software, pp. 231-247, 1995.

[3] Nicolas Anquetil and Timothy C. Lethbridge, "Experiments with Clustering as a Software Remodularization Method", Proceedings of the 6th Working Conference on Reverse Engineering, pp. 235-255, 1999.

[4] 최재훈, 한종진, 박종진, 양재동, "구조적인 시소러스 구축을 지원하는 객체 기반 정보 검색 모델", 정보과학회논문지, 제24권 제11호, 1997.