컴포넌트 검색을 위한 퍼지 시소러스의 성능 평가

채 은 주, 한 정 수*, 김 귀 정**
천안대학교, 천안대학교*, 건양대학교**

Performance Evaluation of Fuzzy Thesaurus for Component Retrieval

Chae eun-ju, Han jung-soo*, Kim gui-jung**
Division of Info. & Comm. Cheonan Univ*. Division of IT Konyang Univ.**
E-mail :cobi80@cheonan.ac.kr, jshan@cheonan.ac.kr, gjkim@konyang.ac.kr

요 약

본 논문은 질의 확장을 통한 퍼지 시소러스와 기존의 시소러스 그리고 직접 매칭 검색 등을 시뮬레이션을 통하여 재현율과 정확도를 통하여 성능평가 하였다. 실험은 임계치를 이용한 평가, 질의 확장을 이용한 평가, 재사용 만족도를 통하여 실험하였다. 실험 결과 퍼지 시소러스의 컴포넌트 검색 효율성이 뛰어남을 증명할 수 있었다.

Abstract

This paper compared fuzzy thesaurus through query extension with existent thesaurus and direct matching retrieval etc. Performance of fuzzy thesaurus is evaluated by the recall and precision through simulation. An experiment did through assessment that use critical value, query extension, and reusability satisfaction. As a result of the experiment, we knew component retrieval efficiency of fuzzy thesaurus excels.

I. 서론

객체지향 방법론의 영향으로 객체의 효율적 관리와 사용 방법에 관한 관심이 높아지고 있으며 이를 활용하기 위해서는 실제적인 개념과 상황에 따른 올바른 객체의 선택과 적용이 필요하다. 소프트웨어의 효율적인 개발을 위해서는 기존에 개발된 재사용 가능한 컴포넌트들을 효율적으로 분류하여 라이브러리에 구조적으로 저장하는 방법, 사용자의 요구사항을 만족하는 컴포넌트를 신속, 정확하게 검색하는 방법, 사용자의 요구사항에 따라 검색한 컴포넌트의 이해를 위한 컴포넌트의 표현 방법, 새로운 소프트웨어 시스템에 결합시키는 방법 등이 개발되어야 한다[1].

기존의 시소러스 관련 연구를 살펴보면 먼저 ACM의 CRCS(Computing Reviews Classification Structure)[2], FIRMS[3], 계층적 시소러스 시스템[4] 등이 있으며, 본 연구에서는 본 연구에서 개발한 퍼지 시소러스의 성능평가를 위하여 기존의 시소러스, 직접 매칭을 통한 방법 등과 비교하였다. 비교 방법은 시뮬레이션을 통하여 임계값을 이용한 평가, 임계값에 따른 재현율과 정확도를 이용한 평가, 질의 확장에 따른 성능평가, 컴포넌트 재

사용 만족도를 통하여 평가하였다. 평가 결과 퍼지 시소러스의 효율성이 22.3% 증가하였음을 증명하였다. 또한 질의확장을 통한 검색이 25%의 선택의 폭을 넓혀준 결과를 가져왔다. 따라서 본 연구는 퍼지 시소러스의 검색 효율을 증명하였다.

II. 시뮬레이션 환경

퍼지 시소러스의 성능 평가를 위한 질의 확장 과정과 질의 확장의 임계치 설정을 위한 시뮬레이션에 대해서 기술한다. 시뮬레이션에 사용된 컴포넌트는 Visual C++ Class Library로 구성하였다. 이 클래스들은 범용 라이브러리이며, 특정 범주에 치우쳐 포함되지 않으므로, 응용 개발을 위하여 일반적인 수준에서의 다양한 클래스를 제공해 줄 수 있다. 또한 이 라이브러리 내에 있는 클래스의 이름은 일반적인 명명 규칙을 따르고 있기 때문에 개발자로 하여금 쉽게 클래스를 구별할 수 있도록 해준다. 본 연구에서 사용한 시뮬레이션 환경이 표1에 나타나 있다. 총 131개의 컴포넌트가 11개의 Class Concept Category(CCC)에 포함되어 있으며, 각 CCC는 컴포넌트에 따라 최소 1개에서 최대 22개까지의 클래스로 구성되어 있다. 컴포넌트에 나타난 CCC의 평균 클래스 수는 약 8개이고, 중복을 포함하여 총 1023개의 클래스가 존재하며 독립된 303개의 클래스로 구성되어 있다.

표 1 시뮬레이션 환경

CCC / 특징	All	Service	Interface	Window	Structure	Network
컴포넌트 수	131	43	37	55	42	27
CCC 내의 최소 클래스 수	1	1	1	1	1	1
CCC 내의 최대 클래스 수	22	23	19	15	22	11
평균 클래스 수	7.8	8.8	6.2	8.0	7.5	5.7
CCC 내의 전체 클래스 수	1023	377	299	442	314	154

CCC / 특징	Exception	OLE	Frame	Processing	Graphic	Interactive
컴포넌트 수	41	23	42	32	49	18
CCC 내의최소 클래스 수	2	1	1	1	1	1
CCC 내의최대 클래스 수	11	8	9	13	22	12
평균 클래스수	5.0	3.2	4.1	6.1	12.7	6.0
CCC 내의전체 클래스 수	203	74	172	197	620	108

III. 퍼지 시소러스 성능 평가

1.질의 확장 임계치 평가

검색 시스템의 시소러스 효율성은 질의에 대한 효과적인 확장에 달려 있다. 본 연구에서는 컴포넌트 검색의 재현율을 최대한 보장해줄 수 있는 최적의 질의 확장 임계치를 시뮬레이션을 통하여 설정하였다. 이를 위해 시소러스 내에서 임의의 클래스 10개를 선택하여 클래스에 대한 유사 확장 집합(similar expended sets)을 선정하였다. 10개의 클래스에 대해 확장된 질의와 유사 확장 집합과의 비교를 통하여 정확도와 재현율을 측정하였다. 이때 질의 확장에 있어 유의값을 0.6에서부터 1.0까지 0.05의 간격으로 각각의 정확도와 재현율을 측정하였다. 정확도와 재현율을 측정하는 방법은 다음과 같다.

$$\text{재현률} = \frac{\text{확장된 유의어 중 유사확장 집합에 속한 유의어의 수}}{\text{유사확장 집합의 수}}$$

$$\text{정확도} = \frac{\text{확장된 유의어 중 유사확장 집합에 속한 유의어의 수}}{\text{확장된 유의어의 수}}$$

표 2. 임계값 검색 효율

임계치	검색 효율	
	정확도	재현율
0.60	0.287	0.853
0.65	0.323	0.853
0.70	0.603	0.811
0.75	0.620	0.500
0.80	0.734	0.428
0.85	0.812	0.329
0.90	0.829	0.285
0.95	0.860	0.285
1.00	0.870	0.210

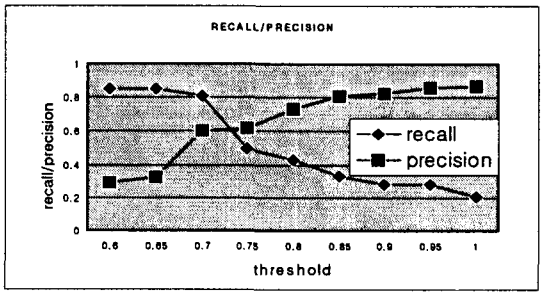

그림 1. 임계값에 따른 정확도/재현율 비교

표2는 임의로 추출한 10개 클래스에 대한 임계값 별 검색 효율을 나타낸 것이고, 그림1은 정확도와 재현율의 평균을 임계값에 따라 그래프로 보여준다. 임계값이 높아질수록 정확도가 좋아지고,

낮아질수록 재현율이 향상됨을 알 수 있다. 그러나 임계값 0.7 미만이 되면 정확도가 0에 가깝게 되어 검색 효율이 떨어지고, 임계값 0.8부터는 0.7과 정확도에 있어서는 별 차이가 나지 않지만 재현율은 상당히 떨어짐을 알 수가 있다. 따라서 본 연구에서는 정확도를 유지하면서 재현율을 최대한 보장할 수 있는 범위인 임계값 0.7 이상을 질의 확장의 범위로 설정하였다.

2. 질의 확장에 따른 성능 평가

임계값 0.7이상의 질의 확장과 이에 따른 시소러스를 성능 평가하였다. 방법은 하나의 질의에 대하여 5번의 확장을 시행한 후, 정확도와 재현율의 변화를 비교하는 것이다. 임계값 0.7 이상인 모든 클래스를 질의에 포함시키고, 새롭게 질의에 포함된 클래스에 대해서 다시 0.7 이상인 클래스를 질의에 포함시키는 방법으로 총 5번의 질의 확장을 시행한다. 이는 확장 과정에 따라 질의에 대한 재현율과 정확도의 변화를 알기 위함이다. 표3에서처럼 "Window"를 5번 확장한 결과 모두 8 개의 질의로 확장되었다. 10개의 클래스에 대해 시행한 결과의 정확도와 재현율 비교가 표4와 그림2에서 보여준다. 확장이 진행될수록 정확도가 떨어지고, 재현율이 향상됨을 알 수 있다. 그러나 확장이 한번도 이루어지지 않은 경우에는 정확도는 높지만, 재현율이 낮아 검색 효율이 떨어지며, 확장이 3번 이상 이루어진 경우에는 재현율에 있어서는 별 차이가 나지 않지만 정확도가 상당히 떨어짐을 알 수가 있다.

표 3. "Window" 클래스 확장 과정

확장 과정	질의
Q1	**Window**
Q2	*Window* **View : 0.84** ***AnimateCtrl : 0.79***
Q3	Window View **FrameView : 0.93** **ScrollView : 0.77** **CtrlView : 0.85** AnimateCtrl
Q4	Window View FrameView CtrlView ScrollView **RecordView : 0.80** AnimateCtrl
Q5	Window View FrameView CtrlView ScrollView RecordView **DaoRecordView : 0.92** AnimateCtrl

표 4. 확장 과정의 검색 효율

확장 과정	검색 효율	
	정확도	재현율
Q1	0.821	0.265
Q2	0.598	0.793
Q3	0.234	0.834
Q4	0.205	0.849
Q5	0.156	0.856

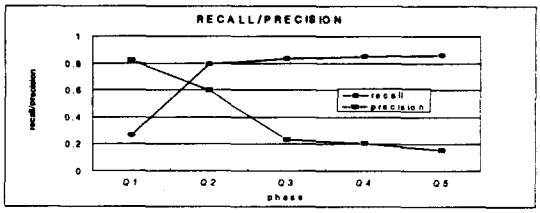

그림 2. 확장에 따른 정확도/재현율 비교

3. 제안한 시스템 효율

시소러스에 의한 검색 시스템은 다양한 후보 컴포넌트를 검색할 수 있고, 정확한 컴포넌트가 존재하지 않을 경우 질의 확장을 통하여 검색의 재현율을 높여 재사용성을 향상시킬 수 있다는 장점이 있다. 따라서 본 연구에서는 시소러스의 효율성을 평가하기 위하여 검색 효율성의 기준이 되는 재현율과 정확도를 측정하였다. 질의는 임의로 20개를 선정하였고, 시소러스를 사용하지 않은 것과 본 연구에서 제안한 방법으로 시소러스를 사용한 경우의 재현율을 0.1 단위로 변화시키면서 정확도의 변화를 측정한 후, 정확도의 평균을 비교하였다. 표5에서처럼 본 연구에서 제안한 시소러스는 시소러스를 사용하지 않은 검색에 비해 효율성에 있어서 22.3%((0.769-0.629)/0.629)*100) 정도 크게 향상되었음을 보여주고 있다. 검색 효율의 차이는 질의 확장에 의해 용어 불일치를 해결하였으며, 유사도 계산에 의해 가장 적절한 컴포넌트가 우선적으로 검색될 수 있다.

표6은 컴포넌트 재사용의 만족도를 보여주고 있으며 이 실험은 임의 질의의 검색 결과에 대하여 재사용된 컴포넌트가 질의를 직접 포함한 컴포넌트인지, 질의 확장에 의해 검색된 컴포넌트인지를 구별하여 비율을 산출하였다. 질의를 직접 포함한 컴포넌트는 패턴 매칭에 의하여 검색되는 결과로 간주할 수 있기 때문에 패턴 매칭의 결과와 비교해보면 질의 확장에 의한 검색은 패턴 매칭 결과에 질의 확장에 의한 컴포넌트까지 검색할 수 있다는 의미를 가지고 있다. 따라서 시소러스에 의

한 검색은 질의에 대한 후보 컴포넌트까지 검색되어 질의를 직접 포함하는 컴포넌트의 검색보다 선택의 폭이 25% 더 넓다는 결과를 보여주고 있다.

표 5. 재현율과 정확도의 비율

Recall	Not 시소러스 경우의 Precision	제안한 시소러스 경우의 Precision
0.1	0.68	1.00
0.2	0.70	0.95
	• • • •	
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%

표 6. 컴포넌트 재사용 만족도

	A		B		C		D		E	
컴포넌트 수(비율)	12		4		8		0.9		0.1	
	13		4		9		0.8		0.2	
	9		1		9		0.3		0.7	
	10		5		5		0.9		0.1	
	8		1		7		0.6		0.4	
	17		6		11		0.8		0.2	
	11		5		6		0.8		0.2	
	7		2		5		0.9		0.1	
평균	10.88		3.5		7.5		0.75 (평균비율)		0.25 (평균비율)	
유사도순	1	2	3	4	5	6	7	8	9	10
평균 재사용성	31.1	30	21	3	4.5	3.5	5	0	0.5	1.4

A :전체 검색된 컴포넌트 수
B :질의를 직접 포함한 컴포넌트 수(패턴매칭)
C :질의 확장에 의해 검색된 컴포넌트 수
D :질의를 포함한 컴포넌트 선택비율(패턴매칭)
E :질의 확장으로 검색된 컴포넌트의 선택 비율

Ⅳ. 기존 시소러스와의 성능비교

표7에서와 같이 본 연구에서 제안한 시소러스와 기존의 시소러스와 비교 분석하였다. 본 연구의 시소러스는 컴포넌트를 검색 대상으로 하고, 기본 구성은 클래스로 이루어져 있다. 또한 클래스를 기능별로 구분하여 클래스간 연관성을 주기 위해 CCC로 분류하였다. 질의 연산은 질의 중요도와 컴포넌트 별 가중치를 이용하여 불리언 연산을 시행함으로써 요구사항을 정확히 표현할 수 있었다. 또한 통계적 방법을 이용하여 클래스 빈도와 가중치를 계산하고 이를 퍼지 시소러스로 구축함으로써 클래스에 대한 질의 확장을 효율적으로 수행하도록 하였다. 기존의 타 시스템의 시소러스와 비교해 보면 검색 대상이 함수, 문서 또는 클래스로 국한되기 때문에 진정한 의미의 객체지향 컴포넌트의 검색이라고 볼 수 없다. 또한, 범주 분류에 있어서도 대부분의 경우 수작업을 통하여 항목을 분류해야 하지만, 본 연구에서 제안한 CCC는 클래스 상속관계를 기본으로 분류되기 때문에 그 의미가 자연스럽고 클래스 간의 기능 분류가 자동으로 이루어 질 수 있다는 장점을 가지고 있다. 특히 계층적 시소러스는 상속관계를 고려한 객체지향 시소러스이긴 하지만, 코드에서 추출한 함수명과 인수를 가지고 시소러스를 구축하기 때문에 데이터양이 너무 방대해져 구축과 관리, 갱신에 어려움이 따른다. 객체기반 시소러스와 구조적 시소러스의 경우에도 초기 개념 설정이 모두 수작업으로 이루어져 이에 따라 시소러스의 성능이 크게 좌우되기 때문에 전문가의 주관에 따라 시소러스가 구축될 가능성을 많다.

표 7. 시소러스 비교 분석

항목 / 시스템	검색대상	구성항목	범주분류	질의연산 방법	시소러스 구축 방법
No Thesaurus	함수 클래스	함수형 데이터형	단순 category	string matching	×
계층적 시소러스[4]	클래스	함수명 인자명	context	context 조합	퍼지시소러스 +통계
제안한 시소러스	컴포넌트	클래스	CCC	불리언 연산	퍼지시소러스 +통계 +CCC상속

Ⅴ. 결론

본 연구는 퍼지 시소러스와 기존의 시소러스를 시뮬레이션을 통하여 성능평가 하였다. 퍼지 시소러스의 성능평가를 위하여 기존의 시소러스, 직접 매칭을 통한 방법 등과 비교하였다. 비교 방법은 시뮬레이션을 통하여 임계값을 이용한 평가, 임계값에 따른 재현율과 정확도를 이용한 평가, 질의 확장에 따른 성능평가, 컴포넌트 재사용 만족도를 통하여 평가하였다. 평가 결과 퍼지 시소러스의 효율성이 22.3% 증가하였음을 증명하였다. 또한 질의확장을 통한 검색이 25%의 선택의 폭을 넓혀준 결과를 가져왔다. 따라서 본 연구는 퍼지 시소러스의 검색 성능이 높음을 증명할 수 있었다.

참 고 문 헌

[1] R.Mili, A.Mili, and R.T.Mittermeir, "Storing and retrieving software components : a refinement based System," IEEE Transaction on Software Engineering Vol.23, No.7, pp. 445-460, 1997.

[2]ACM, "The Full Computing Reviews Classification System," ACM, New York, 1992.

[3] P. Subtil, N. Mouaddib and O. Foucaut, "A Fuzzy Information Retrieval and Management System and Its Applications," The Proceeding of the ACM Symposium on Applied Computing, pp.537-541. Feb. 1996.

[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.