

컴포넌트 재사용을 위한 효율적인 사용자 검색 피드백에 관한 연구

한정수^{1*}

A Study on Efficient User Retrieval Feedback for Component Reuse

Jung-Soo Han^{1*}

요약 본 논문은 컴포넌트 검색 성능을 향상시키기 위해 사용자 피드백을 효율적으로 수행하는 방법을 제안하고자 한다. 기존의 퍼지 기법이 적용된 피드백의 단점을 극복하기 위하여 검색된 컴포넌트의 선택여부에 따라 동일한 함수이지만 학습률을 달리할 수 있는 가우시안 함수를 이용한 상호작용 함수를 제안한다. 또한 피드백 함수에 사용자 프로필을 적용하여 사용자 등급에 따라 사용자 의견이 시스템에 반영되는 정도를 다르게 해주었다. 본 연구의 사용자 검색 피드백은 가우시안 함수를 적용한 피드백 함수와 사용자 프로필을 이용하여 시스템을 장기간에 걸쳐 서서히 변화시킬 수 있는 적응형 검색 방법이다.

Abstract The paper describes a method of user feedback in order to enhance the retrieval effectiveness. In this paper, to overcome a weak point of the existing feedback function adapting fuzzy technique, we proposed the interaction function using gaussian function that gives different learning rate according to choice of components with same function. And, we grade degree that the user opinion is reflected to a system by applying user profile to the feedback function. User retrieval feedback method is adaptive retrieval method that makes a slow change for a long time using feedback function adapting gaussian function and user profile.

Key Words : 사용자 피드백, 사용자 프로필, 컴포넌트 검색, 가우시안 함수

1. 서론

효율적인 컴포넌트 재사용을 위해서는 재사용을 편리하게 해주는 컴포넌트 분류 방법과 검색 기술이 필요하다. 기존의 퍼지 기반의 컴포넌트 검색의 경우 클래스 개념 범주별 출현빈도에 따라 시소러스를 구축해야하고 클래스의 수가 적은 도메인이나 특정 범주에 클래스가 몰려있는 환경에서는 효율성이 떨어지는 경향이 있다. 또한 검색 결과는 유사도에 의존하기 때문에 컴포넌트들에 대한 정확한 기능 정의에만 한정되어 있어 더 많은 컴포넌트의 이해를 위한 정보가 요구된다. 따라서 본 논문에서는 퍼지값을 적용하는 방법을 사용하면서 컴포넌트의 효율적인 재사용을 위해 후보 컴포넌트들을 사용자 의견에 따라 우선순위로 검색하는 방법을 제안한다. 컴포넌트 검색을 위하여 컴포넌트와 질의 사이의 신뢰값과 피드백

함수를 이용한다[1][2]. 신뢰값은 컴포넌트와 질의의 일치정도를 퍼지로 표현하며, 피드백 함수는 사용자의 등급을 나타내는 사용자 프로필 값을 이용하여 구축된다.

기존의 피드백 함수는 컴포넌트 선택 횟수에 따라 퍼지값을 적용하는 방법이 주로 사용되고 있는데, 이러한 방법은 몇 가지 문제점을 가지고 있다[2][3]. 첫째, 파라메타의 선택에 따라 함수의 성능이 크게 좌우된다. 파라메타들을 임의로 초기화 시켜줘야 하고, 그 값에 따라 시스템의 성능에 커다란 영향을 미친다. 둘째, 시스템의 크기가 어느 정도 이상 커야하며 100회 이상의 시행이 이루어져야 안정적인 결과를 기대할 수 있다. 셋째, 컴포넌트를 선택할 때마다 매번 4가지 경우의 그래프를 재구성해야 하는 어려움이 있다. 그러므로 가중치나 파라메타의 임의 설정을 최소화하고, 구축 절차를 간소화하여 사용자와 시스템 간에 적응적·동적으로 상호작용 할 수 있는 피드백 함수의 생성이 매우 요구된다. 본 논문에서는 위에서 열거한 기존의 피드백 단점을 극복하기 위하여 검색

¹백석대학교 정보통신학부

*교신저자: 한정수(jshan@bu.ac.kr)

색된 컴포넌트의 선택여부에 따라 동일한 함수이지만 학습률을 달리할 수 있는 가우시안 함수를 이용한 동적 상호작용 함수를 제안한다.

또한 본 논문에서는 사용자 등급의 척도로서 사용자가 가지고 있는 기술을 사용자 프로필로 정의하였다. 이 프로필을 벡터값으로 표현하였고, 각 벡터 요소를 사용자의 도메인 기술(domain skill), 작업 기술(task skill), 그리고 전략 기술(strategy skill)로 정의하였다. 피드백 함수는 어떤 등급의 사용자가 어떤 컴포넌트를 선택하느냐에 따라서 그 값이 서로 다르게 수정된다. 이 방법은 단일 질의에 대한 최적화보다는 시스템의 전반적인 향상을 목적으로 한다. 본 연구에서 제안한 검색 피드백 방법은 가우시안 함수와 사용자 프로필을 적용한 피드백 함수를 이용하여 시스템을 장기간에 걸쳐 서서히 변화시킬 수 있도록 하였다.

2. 기존 연구

피드백 함수는 사용자가 어떤 컴포넌트를 선택하느냐에 따라서 그 모양이 수정될 수 있다. 사용자 검색 환경에 따른 컴포넌트의 검색순위 재구성이 이루어지게 된다. 퍼지를 기반으로 한 피드백 함수(D)의 수정 방법은 다음과 같다[3]. 우선순위에 따라 검색된 컴포넌트 중 사용자가 첫 번째 컴포넌트를 선택하지 않고 k번째 컴포넌트를 선택했을 때, 첫 번째부터 k번째 컴포넌트의 피드백 함수는 아래와 같은 식에 의해 변환된다.

$$D_{new}(t) = (1 - \delta)D_{old}(t) + \delta D_{corr}(t) \quad (1)$$

여기에서, 첫 번째부터 (k-1)번째 컴포넌트의 경우에 $D_{corr}(t)$ 는 다음과 같이 수정된다.

$$\begin{aligned} D_{corr}(t) &= (1 + \gamma)t - \gamma, & t \leq 0.5 \\ D_{corr}(t) &= 2(1 + \gamma)(1 - t) - \gamma, & t > 0.5 \end{aligned} \quad (2)$$

그리고, 사용자가 선택한 k번째 컴포넌트의 $D_{corr}(t)$ 는 다음과 같이 수정된다.

$$\begin{aligned} D_{corr}(t) &= 2(1 - \gamma)t + \gamma, & t \leq 0.5 \\ D_{corr}(t) &= (1 - \gamma)(1 - t) + \gamma, & t > 0.5 \end{aligned} \quad (3)$$

파라메타 γ 는 애플리케이션 기술자에 의해 조절되며, 시스템의 변화정도를 나타낸다. γ 의 값이 크면 클수록, 함수의 모양이 더 급격히 변하게 되어 사용자 피드백의 결과가 시스템에 더 빨리 반영된다. 파라메타 δ 는 시스템 반영 정도를 다르게 해주는 역할을 한다. 그러나 이 방법은 γ 에 따라 값의 변동이 너무 크게 나타나고, 컴포넌트를 선택할 때마다 매번 4가지 타입의 그래프를 재구성해야 한다.

3. 사용자 검색 피드백 함수

3.1 가우시안 함수

본 연구는 사용자 집단의 요구에 동적으로 반응할 수 있는 피드백 함수를 제안하고자 한다. 기존의 피드백 단점을 극복하기 위하여 검색된 컴포넌트의 선택여부에 따라 동일한 함수이지만 학습률을 달리할 수 있는 가우시안 함수를 이용한 동적 상호작용 함수를 제안한다. 가우시안 함수를 피드백 함수로 채택한 이유는 다음 세 가지의 장점을 가우시안 함수가 가지고 있기 때문이다.

- ① 좌우대칭인 함수를 이용함으로써 매번 여러 가지의 그래프를 재구성하는 번거로움을 피할 수 있다.
- ② 1부터 0 사이의 값으로 한정시킬 수 있다.
- ③ 최대값(1)과 최소값(0)은 변하지 않으면서 폭을 조절함으로써 서로 다른 값을 생성할 수 있다.

본 연구의 사용자 피드백은 함수의 기울기를 변화시킴으로써 이루어진다. 즉, 기존의 방법은 경우에 따라 매번 다른 함수를 이용하거나 가중치를 직접 변경시키는 방법이지만, 제안한 방법은 하나의 함수를 이용하여 다양한 값을 생성할 수 있다. 본 연구에서 제안한 피드백 함수식은 다음과 같다.

$$D_{corr}(t) = e^{-\frac{(H + (-1)^n \frac{C_i}{T}) \cdot t^2}{2}} \quad (4)$$

T : 누적 검색 횟수
 C_i : i 번째 컴포넌트가 검색된 횟수
 H : 함수의 폭
 n : 0 or 1

파라메타 H에 의해 함수의 폭이 결정된다. H가 커질수록 폭은 작아지고 $D_{corr}(t)$ 값은 감소하며, H가 작아질수록 폭은 커지고 $D_{corr}(t)$ 값은 증가한다. 이 식은 가우시안 함수의 폭 H를 감소·증대시켜 각기 다른 학습률

을 컴포넌트에 적용시키기 위함이다. 사용자가 k번째 컴포넌트를 선택했을 때, 첫 번째부터 (k-1)번째 컴포넌트에는 기존의 값보다 작은 값을 부여하기 위해 n=0을 설정하여 폭을 감소시킨다. 또한 선택된 k번째 컴포넌트에는 더 높은 신뢰값을 주기 위하여 n=1을 설정하여 폭을 증대시킨다.

가우시안 함수는 파라메타 H에 따라 성능에 커다란 차이를 보인다. 본 연구에서는 실험을 통해 파라메타에 따른 성능을 비교·분석하고, 이를 토대로 가장 효율적인 동적 함수를 생성하였다. 상호작용 함수로 사용된 가우시안 함수의 성질을 분석하는 것은 검색 피드백의 성능을 향상시키기 위한 절차로써 파라메타들의 변화와 그로 인한 피드백 함수의 성능을 알아 볼 수 있다. 그림 1은 H에 따른 가우시안 피드백 함수의 모양이다. H가 증가함에 따라 그래프의 폭이 줄어드는 것을 알 수 있다.

이렇게 다양한 파라메타의 선택에 따라 피드백 함수의 성능은 크게 좌우될 수 있다. 따라서, 본 연구에서는 사용자 피드백을 가장 효과적으로 반영할 수 있는 가우시안 함수를 생성하기 위하여 파라메타에 따른 성능을 비교·분석하였다. 파라메타 H의 변화로 인한 피드백 함수의 성능을 식 (5)를 이용하여 시간영역에 대해서 평가하였다. 이 식은 신경망의 학습정도에 대한 오차를 구하기 위하여 사용되는 식에 기반을 둔다[4]. 여기에서 D_i 는 피지 함수값을 나타내며, μ_i 는 가장 이상적인 수렴점을 나타낸다. 그림 2는 파라메타 H의 변화에 따른 함수의 오차를 나타낸 것이다. 그림에서 알 수 있듯이 H가 커질수록 오차율이 작아져 성능이 향상되지만, H가 일정한 임계치를 넘으면 오히려 성능이 떨어짐을 알 수 있다.

$$error(t) = \sum_i [D_i(t) - \mu_i]^2 + (D_i(t) - \mu_i)^2 \quad (5)$$

본 연구에서는 파라메타에 따른 성능을 비교·분석한 결과 H가 7일 경우에 가장 효과적인 피드백 함수가 생성됨을 알 수 있었다. 따라서, 본 연구에서 제한한 컴포넌트 검색에서의 사용자 피드백을 가장 효율적으로 반영할 수 있는 가우시안 피드백 함수식은 다음과 같이 정의할 수 있다.

$$D_{corr}(t) = e^{-\frac{(7+(-1)^n \frac{C_i}{T}) \cdot t^2}{2}} \quad (6)$$

위의 식을 이용하여 첫 번째부터 k번째 컴포넌트의 피드백 함수는 아래 식에 의해 변환된다[5].

$$D_{new}(t) = (1 - \beta)D_{old}(t) + \beta D_{corr}(t) \quad (7)$$

파라메타 β 는 사용자 등급에 따라 시스템 반영 정도를 다르게 해주는 역할을 한다.

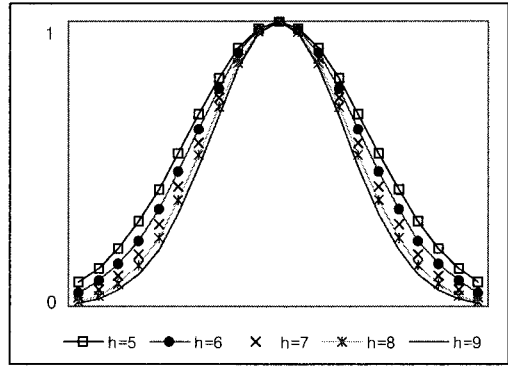


그림 1. 파라메타 H에 따른 가우시안 함수

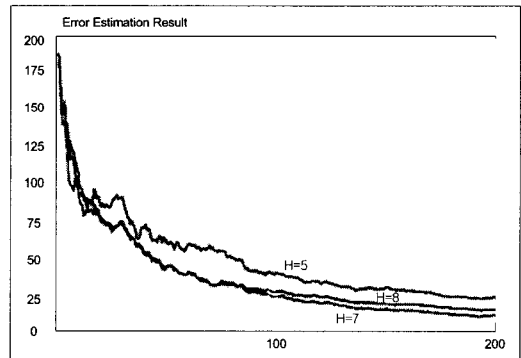


그림 2. 파라메타 H에 따른 오차율

3.2 사용자 프로필의 정의

사용자 등급을 표현하는 β 의 값을 정의하기 위해서는 사용자의 개념을 좀더 상세히 세분화할 필요가 있다. 소프트웨어 개발 환경에서 기술자와 개발자는 모두 각각의 고유한 위치와 책임이 주워지게 되며, 이에 따라 재사용 가능한 컴포넌트의 선택이 달라질 수 있다. 이에 본 논문에서는 사용자 등급의 척도로서 사용자가 가지고 있는 기술을 사용자 프로필로 나타냈다[6]. 이 프로필은 벡터값 $\{a_1, a_2, a_3\}$ 으로 표현되고 0에서 1사이의 값을 갖는다. 각 벡터 요소는 사용자의 도메인 기술(domain skill), 작업 기술(task skill), 그리고 전략 기술(strategy skill)을 나타낸다. 작업 기술은 사용자가 참가하고 있는 개발 그룹의 수를 말하며, 도메인 기술은 특정 응용 도메인에 참가한 사용자의 프로젝트 수를 말한다. 전략 기술은 팀 내에서 차지하는 사용자의 책임 정도치를 나타낸

다. 시스템에 접속할 때마다 사용자 프로필이 계산되고, 이는 피드백 함수 $D_{new}(t)$ 에서 β 를 계산하기 위해 사용된다. 사용자 프로필에 기술 가중치(WA)를 적용함으로써 사용자 등급 β 값을 구할 수 있다. 다음은 파라미터 β 를 구하는 식이다.

$$\beta = M \sum_{i=1}^3 w_i \alpha_i \quad (8)$$

제약조건은 $\sum_{i=1}^3 w_i = 1$ 이고, M 은 $\beta \ll 1$ 를 만족하는 최대값이다. 본 논문에서는 $M=0.1$ 로 하였다. 예를 들어, 6명의 전문가와 12명의 초급자, 그리고 이들을 관리하는 2명의 관리자로서 이루어진 개발팀이 있다고 하자. 이때, 관리자는 높은 전략 기술과 도메인 기술을 가지고 있지만 작업 기술에는 제한이 있을 것이다. 그러므로 프로필을 벡터값으로 표현할 때, 관리자의 프로필은 $\{\alpha_1=0.8, \alpha_2=0.1, \alpha_3=0.8\}$ 로 표현될 수 있다. 반면, 초급 프로그래머는 좋은 작업 기술은 있지만, 도메인과 전략에 대한 주요 지식은 많이 부족할 것이다. 그러므로 초급자의 프로필은 $\{\alpha_1=0.1, \alpha_2=0.8, \alpha_3=0.1\}$ 로 표현될 수 있다. 또한 고급 기술자는 도메인 기술이 가장 뛰어나고 적당한 작업 능력을 가지고 있으며, 전략에 대한 지식은 부족할 것이다. 그러므로 고급자의 프로필은 $\{\alpha_1=0.8, \alpha_2=0.6, \alpha_3=0.2\}$ 로 표현될 수 있다.

가중치 w_i 는 시스템을 사용하는 개발팀의 각 기술자들이 피드백 과정에 어느 정도 역할을 담당할지 결정한다. 관리자가 선택한 컴포넌트와 초급 기술자가 선택한 컴포넌트가 재사용에서 같은 정도의 중요성을 갖는다고 볼 수는 없을 것이다. 이를 위해 사용자 프로필에 기술 가중치를 적용하는 것이다. 이와 같은 방법은 재사용 개발환경에서 어떤 컴포넌트가 정책적으로 가장 유용할 것인가를 잘 알고 있는 관리자의 의견을 충분히 반영하고자 하는데 그 목적이 있다. 본 연구에서는 사용자 프로필 벡터의 가중치 값을 $w_1=0.3, w_2=0.2, w_3=0.4$ 와 같이 설정하였다[6]. 전략 기술에 가장 큰 가중치를 부여하여 관리자의 의견이 시스템 피드백에 가장 잘 반영될 수 있도록 하였다. 사용자 프로필에 기술 가중치를 적용하여 사용자 등급 β 값을 구하는 식은 다음과 같다.

$$\begin{aligned} \text{관리자에 대한 사용자 등급 } \beta &= M \sum_{i=1}^3 w_i \alpha_i \\ &= 0.1 \times (\{0.8, 0.1, 0.8\} \times \{0.3, 0.2, 0.4\}) = 0.058 \end{aligned}$$

$$\begin{aligned} \text{고급자에 대한 사용자 등급 } \beta & \\ &= 0.1 \times (\{0.8, 0.6, 0.2\} \times \{0.3, 0.2, 0.4\}) = 0.044 \end{aligned}$$

$$\begin{aligned} \text{초급자에 대한 사용자 등급 } \beta & \\ &= 0.1 \times (\{0.1, 0.8, 0.1\} \times \{0.3, 0.2, 0.4\}) = 0.023 \end{aligned}$$

4. 컴포넌트 검색

컴포넌트 검색을 위한 질의와 컴포넌트의 신뢰도 계산 과정은 다음과 같다[7].

- ① 질의와 컴포넌트간의 동치관계 계산.

$$\begin{aligned} Eq(Query(u), Comp(v)) & \\ &= SYNON(Query(u), Comp(v)) \end{aligned} \quad (9)$$

u : 질의에 있는 질의어 갯수
 v : 컴포넌트에 있는 클래스 갯수

- ② 질의와 컴포넌트간의 함축관계 계산.

$$\begin{aligned} Imp(Query(u), Comp(v)) & \\ &= \max \{w(Query(u)), w(Comp(v))\} [Eq(u, v)] \end{aligned} \quad (10)$$

- ③ 질의와 컴포넌트 클래스의 만족도 계산.

$$Sat(Query, Comp(v)) = \frac{[\sum_{i=1}^U Imp(u, v) \times Eq(u, v)]}{U} \quad (11)$$

- ④ 질의와 컴포넌트간의 유사도 계산.

$$Sim(Query, Comp) = Sat \times W \quad (12)$$

- ⑤ 신뢰값(confidence value:CV) 계산.

$$CV = L \sum_{i=1}^k D_i CV_i \quad (13)$$

신뢰값은 사용자에 의해 주어진 질의에 대한 최적의 컴포넌트를 검색해주며, 또한 사용자 피드백을 반영한다. CV_1, CV_2, \dots, CV_k 는 유사도 계산에서 도출된 유사값(Sim)을 오름차순으로 정렬한 값이다. 여기에서 L 은 지나치게 작은 값이 나올 경우 그 값을 증폭시키는 역할을 한다. 그림 3은 가우시안 함수를 피드백 함수로 이용했을 때 컴포넌트의 신뢰값 수정방법을 보여준다. 그림에서 CV_1, \dots, CV_k 는 유사값을 오름차순으로 정렬한 값이다. 유사값의 가장 큰 값과 가장 작은 값의 영향을 최소화하고 함수값을 균형있게 적용하기 위함이다.

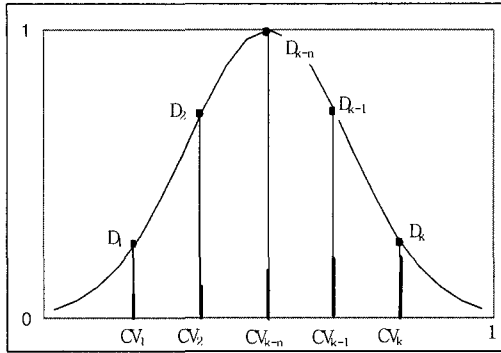


그림 3. 피드백 함수를 이용한 신뢰값 수정

본 연구에서는 제안한 사용자 검색 피드백을 위한 사용자 프로필 정의의 효율성을 다음과 같이 실험하였다. "Document" 컴포넌트, "OleDocument" 컴포넌트, "ClientDocument" 컴포넌트에 대하여 같은 질의를 100번 수행하고 그 CV값의 변화를 실험하였다. 사용자 등급은 관리자, 전문가, 초급자로 나누었으며, 각 등급에 따른 기술 벡터값 $\{a_1, a_2, a_3\}$ 과 이에 따른 β 는 표 1과 같다. 사용자 등급이 높을수록, β 값이 커져 시스템 반영정도가 높아짐을 알 수 있다.

표 1. 사용자 등급 값

구분	사용자	관리자	전문가	초급자
사용자기술값 α	a_1	0.8	0.8	0.1
	a_2	0.1	0.6	0.8
	a_3	0.8	0.2	0.1
가중치 w	w_1	0.3	0.3	0.3
	w_2	0.2	0.2	0.2
	w_3	0.4	0.4	0.4
M		0.1	0.1	0.1
β		0.058	0.044	0.023

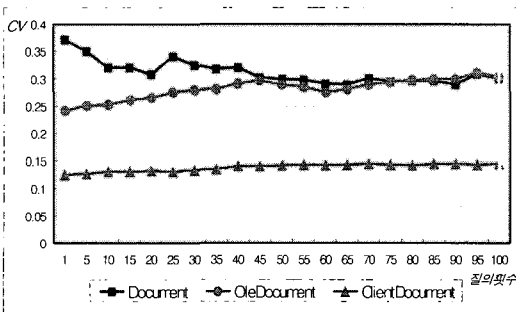


그림 4. 사용자 피드백에 따른 검색

실험에서 100번 질의 중 38번은 관리자에 의해 수행되었으며, 30번은 전문가에 의해 수행되었고 32번은 초급자에 의해 수행되었다. 질의에 대해 사용자에 의해 선택된 컴포넌트의 비율은 각각 "Document" 컴포넌트가 10%, "OleDocument" 컴포넌트가 80%, 그리고 "ClientDocument" 컴포넌트가 10%를 차지하였다. 그림 4는 각 질의에 대한 컴포넌트 선택에 따른 CV의 변화를 나타낸 것이다. 사용자가 "OleDocument" 컴포넌트를 많이 선택함에 따라 "OleDocument" 컴포넌트의 CV값이 점차로 증가함을 볼 수 있다. 이는 초기 CV값은 "Document" 컴포넌트가 더 높지만, 사용자가 "OleDocument" 컴포넌트를 많이 선택함으로써 검색 우선순위가 더 높아짐을 의미한다.

5. 성능평가

본 논문은 컴포넌트 검색 성능을 향상시키기 위해 사용자 피드백을 효율적으로 수행하는 방법을 제안하였다. 제안한 방법은 피드백 함수에 사용자 프로필을 적용하여 사용자 등급에 따라 사용자 의견이 시스템에 반영되는 정도를 다르게 해주었다. 본 연구의 사용자 검색 피드백은 가우시안 함수를 적용한 피드백 함수와 사용자 프로필을 이용하여 시스템을 장기간에 걸쳐 서서히 변화시킬 수 있는 적응형 검색 방법이다. 본 연구에서는 검색 시스템의 효율성을 평가하기 위하여 검색 효율성의 기준이 되는 재현율과 정확도를 측정하였다. 측정 방법은 기존의 방법과 제안한 방법에 의해 검색했을 경우의 검색에 대해서 재현율과 정확도를 비교하였다. 재현율은 전체 컴포넌트 중 적절한 컴포넌트의 수에 대한 검색된 적절한 컴포넌트의 수, 정확도는 검색된 전체 컴포넌트 수에 대한 검색된 적절한 컴포넌트의 수를 이용하여 측정하였다. 질의는 임의로 20개를 선정하였고, 재현율을 0.1 단위로 변화시키면서 정확도의 변화를 측정한 후, 정확도의 평균을 비교하였다[8]. 표 2는 두 경우의 정확도와 재현율의 비율을 나타낸 것이다.

이 실험에서 본 연구에서 제안한 시소러스에 의한 검색은 시소러스를 사용하지 않은 검색에 비해 효율성에 있어서 22.3%(((0.769-0.629)/0.629)*100) 정도 크게 향상되었음을 보여주고 있다. 검색 효율의 차이가 발생하는 가장 큰 이유는 질의 확장에 의해 용어 불일치를 해결하였으며, 유사도 계산에 의해 가장 적절한 컴포넌트가 우선적으로 검색될 수 있다는 데 있다.

표 2. 재현율과 정확도의 비율

Recall	기존의 방법 Precision[2][3]	본연구에서 제안한 경우의 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%

6. 결론

본 논문은 후보 컴포넌트들을 사용자 프로필에 따라 우선순위로 검색할 수 있는 방법을 제안하였다. 피드백 함수로 가우시안 함수를 이용하는데, 가우시안 함수의 폭을 감소·증가시킴으로써 각기 다른 학습률을 컴포넌트에 적용시킬 수 있도록 하였다. 제안한 가우시안 함수는 컴포넌트 선택여부에 따라 폭이 조절되어 각기 다른 학습률을 갖게 되는 동적 함수로써 이용된다. 또한 사용자 프로필을 이용하여 개발팀에 참여하는 사용자들 여러 등급으로 분류함으로써 사용자 피드백이 시스템에 미치는 반영정도를 차별화하는 방법을 제안하였다. 시스템에 접속할 때마다 사용자 프로필이 매번 계산되어지고, 이는 피드백 함수를 계산하기 위해 사용된다.

이와 같이 제안한 사용자 피드백 함수는 사용자와의 동적 상호작용이 가능한 함수로써 사용자 집단의 요구에 능동적으로 반응하여 효율적인 재사용 환경을 제공할 수 있다.

참고문헌

[1] A. M. Zaremski, J. M. Wing, "Signature Matching : A Tool for Using Software Libraries", ACM Transaction Software Engineering and Methodology, Vol. 4, No. 2, pp.146-pp.170, Apr. 1995.

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

[3] E. Damini, M.G.Fugini, "Fuzzy Techniques for Software Reuse", In Proceedings of ACM SIG-APP Conference on Applied Computing, pp.552-557, Feb. 1996.

[4] P.C. Chang. Predictive, Hierarchical and Transform Vector Quantization for Speech Coding. PhD thesis, Stanford University, 1986.

[5] E. Damiani, and M. G. Fugini, "Automatic thesaurus construction supporting fuzzy retrieval of reusable components", Proceeding of ACM SIG-APP Conference on Applied Computing, Feb. 1995.

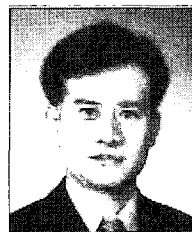
[6] 김귀정, "재사용 환경에서 퍼지 기법을 적용한 사용자 피드백", 한국콘텐츠학회 학술대회논문지, 제2권 제1호, pp.401-405, 5월, 2004.

[7] Gui-Jung Kim, Jung-Soo Han, "Thesaurus Construction using Class Inheritance", Proceedings of the Computational Science and Its Application, LNCS 3482, Springer, pp.748-757, 2005.

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

한 정 수(Jung-Soo Han)

[정회원]



- 1990년 2월 : 경희대학교 전자계산공학과 (공학사)
- 1992년 2월 : 경희대학교 전자계산공학과 (공학석사)
- 2000년 2월 : 경희대학교 전자계산공학과 (공학박사)
- 2001년 3월 ~ 현재 : 백석대학교 정보통신학부 조교수

<관심분야>

소프트웨어공학, CBD, S/W형상관리