

논문 2005-42TE-1-8

컨텍스트 인식 환경에서 레벨화된 정보 검색 기법

(Levelized Information Retrieval Method in Context Awareness Environments)

김 성 립*, 권 준 희**

(Sungrim Kim and Joonhee Kwon)

요 약

컨텍스트 인식 환경에서의 정보 검색 기법은 유비쿼터스 컴퓨팅에서의 핵심 연구 분야 중 하나이다. 컨텍스트 인식 환경에서의 정보 검색 기법에서 정보의 정확성과 신속성은 중요한 고려요소이다. 이를 위해 본 논문에서는 컨텍스트의 레벨화 특성을 이용한 검색 기법을 통해 정보를 정확하고 신속하게 검색하는 새로운 기법을 제안한다. 컨텍스트의 레벨화 특성에 가장 적합한 정보를 검색하기 위해 규칙을 추출한다. 현재 컨텍스트 값과 규칙을 사용하여 가까운 미래에 나타날 컨텍스트 값별 권유 정보를 추출한다. 그리고 접근 점수를 이용하여 가장 가까운 미래에 나타날 레벨화된 컨텍스트 값별 권유 정보를 프리페칭한다. 제안한 기법에서는 컨텍스트의 레벨화 특성을 이용한 권유 정보를 미리 저장함으로써 사용자의 컨텍스트 값이 변경될 때마다 정확하고 신속하게 정보를 검색할 수 있다.

Abstract

The context-aware retrieval method is one of the fundamental characteristics in ubiquitous computing. The essential aims of context-aware retrieval method are retrieving relevant information and delivering information quickly. We propose a new method that retrieves relevant information and delivers information quickly using characteristics of levelized contexts. We extract rules and recommendation information in the near future using context values and rules. Then we prefetch recommendation information in very near future using access score. Our method retrieves relevant information and deliver information quickly by storing only recommendation information to be needed in near future using the characteristics of levelized contexts.

Keywords : Context-awareness, Retrieval method, Ubiquitous computing, Levelized information

I. 서 론

마크 와이저는 유비쿼터스 컴퓨팅을 사람을 포함한 현실 공간에 존재하는 모든 대상물들을 기능적·공간적으로 연결해서 사용자에게 필요한 정보나 서비스를 즉시 제공할 수 있는 기반 기술이라고 정의하였다^[1]. 유비쿼터스 컴퓨팅의 여러 응용 서비스에서 가장 핵심적인

요소 기술 중 하나는 컨텍스트 인식을 통해 각 컨텍스트에 따라 적절한 정보나 서비스를 제공하는 기술이다. 여기서 컨텍스트는 사용자, 공간, 오브젝트 등의 개체와 관련된 모든 정보라고 정의되고, 계속적으로 변화하는 특성이 있다^[2,3].

유비쿼터스 컴퓨팅에서 컨텍스트 인식을 이용한 어플리케이션에 대한 최근 분석에 따르면 미래에 필요한 가장 중요한 핵심 연구 분야 중 하나로 컨텍스트 인식 환경에서의 정보 검색 기법을 들고 있다^[4,5]. 이러한 정보 검색 기법에서 중요하게 고려해야 하는 것은 사용자들에게 필요한 정보를 얼마나 정확하게 검색해 주는가와 검색해야 할 정보의 양이 많은 경우라도 얼마나 신속하게 정보를 검색하는가이다^[4,6].

* 정회원, 서일대학 인터넷정보전공
(Dept. of Internet Info., Seoil College)

** 정회원, 경기대학교 정보과학부 전자계산학 전공
(Dept. of Computer Science, Kyonggi Univ.)

※ 본 논문은 2004년 서일대학 학술연구비의 지원에 의해 연구되었음.

접수일자: 2004년10월27일, 수정완료일: 2004년12월28일

기존의 컨텍스트 인식 검색 기법에서는 컨텍스트의 계속적인 변화에 따라 관심 대상이 되는 모든 상세화된 정보를 검색하므로 검색 시간이 많이 소요되며, 또한 이렇게 검색된 정보가 사용자들에게는 불필요한 정보가 될 수도 있기 때문에 정확한 정보를 검색하는데 문제점이 있다. 따라서 정확하고 신속하게 정보를 검색하기 위해서는 새로운 검색 기법이 요구된다.

이를 위해 본 논문에서는 컨텍스트의 레벨화 특성을 이용한 점진적 검색 기법을 통해 정보를 정확하게 검색하면서도 신속하게 정보를 검색하는 새로운 기법을 제안하고자 한다. 그리고 계속적으로 변화하는 컨텍스트에 대해 레벨화를 수행함으로써 각 컨텍스트에 해당하는 상세한 정보를 한번에 검색하지 않으면서도 컨텍스트 값별로 설정된 레벨에 따라 사용자에게 필요한 정보만을 검색하도록 한다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구들을 살펴보고, III장과 IV장에서 본 논문에서 제안하는 레벨화된 정보 검색 기법의 구성도와 알고리즘을 설명한다. V장에서는 제안된 기법을 실험하고, VI장에서 결론을 맺는다.

II. 관련 연구

컨텍스트 인식 환경에서 정보를 검색하는 기법에 있어 중요한 목표는 필요한 정보만을 얼마나 신속하게 검색하는가이다^[7]. 이러한 목표를 기준으로 기존 연구를 살펴보면 다음과 같다.

첫째, 사용자 프로파일 정보를 현재 컨텍스트와 단순 비교하여 필요한 정보를 검색하는 기법이다. 컨텍스트 환경에서의 대부분의 정보 검색 기법^[4,5]은 사용자가 수동적으로 입력한 사용자 프로파일 정보에 현재 인식된 컨텍스트를 단순히 비교하여 해당 정보를 검색하는 방법이다. 이러한 접근 방법은 컨텍스트에 따라 의미있는 정보를 검색할 때 명시적 데이터에만 의존하므로 필요한 정보를 검색하는데 한계를 보인다.

둘째, 컨텍스트-인식 캐쉬를 이용한 즉각적인 컨텍스트 인식 검색 기법이 있다^[7]. 이 기법은 과거와 미래에 관한 명시적인 컨텍스트 값을 바탕으로 사용자가 미래에 사용하게 될 정보를 미리 캐쉬에 저장하고, 계속적으로 변화하는 컨텍스트에 요구되는 정보를 캐쉬로부터 검색함으로써 신속한 검색 결과를 기대할 수 있다. 그러나 컨텍스트-인식 캐쉬는 미래 정보를 미리 수동적으로 입력해두어야 한다는 문제점과 명시적인 정보에만

의존하고 있어 사용자들에게 필요한 정보를 정확하게 검색해 내지 못한다는 문제점이 있다.

셋째, 컨텍스트 패턴과 멀티 에이전트를 이용한 즉각적인 컨텍스트 인식 검색 기법^[7]이 있다. 이 기법은 컨텍스트 패턴을 추출하고, 추출된 패턴을 사용하여 가까운 미래에 사용하게 될 정보만을 프리패칭하면서도 계속적으로 변화하는 컨텍스트에 유연하게 반응하기 위해 멀티 에이전트 기법을 활용하였다. 그러나 이 기법도 컨텍스트의 변화에 따라 관심 대상이 되는 모든 상세화된 정보를 검색하게 된다. 이는 불필요한 정보 검색과 검색의 신속성에 문제를 발생시킨다.

III. 시스템 설계

1. 시스템 구성

본 논문에서 제안하는 레벨화된 정보 검색 기법의 전체적인 시스템 구성도는 그림 1과 같다.

서버측 구성 요소는 다음과 같다. CCM (Context/Contents Manager)은 컨텍스트 (context)와 콘텐츠 (content)를 관리하는 기능을 수행하고, SA (Server Agent)는 사용자의 변화된 컨텍스트에 해당하는 검색 결과를 클라이언트에 전달하는 기능을 수행한다. LM (Levelizing Manager)은 텍스트의 특성에 따라 레벨화된 정보를 추출하는 기능을 수행한다. 그리고 SM (Selecting Manager)이 이렇게 추출된 결과를 SA에 전달한다.

클라이언트측의 구성 요소는 다음과 같다. RM (Request Manager)과 CM(Context Manager)을 통해 전

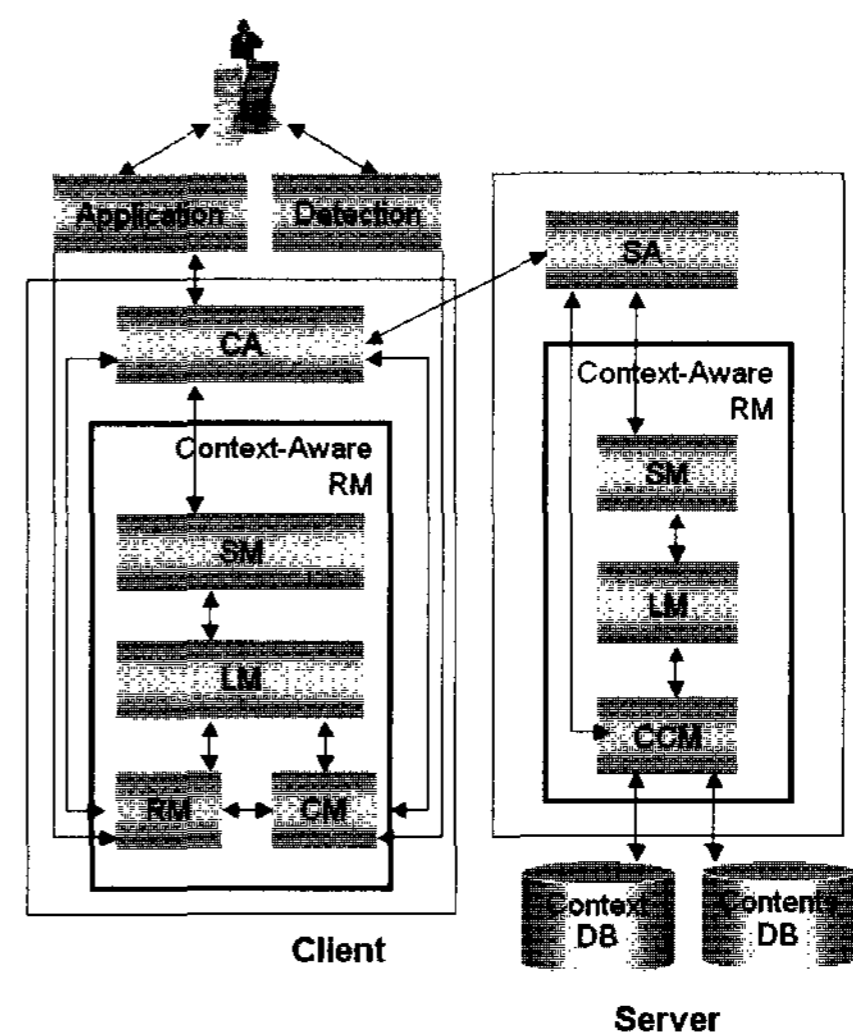


그림 1. 시스템 구성도

Fig. 1. System Architecture.

달받은 정보와 서버로부터 전달받은 정보를 이용해 LM, SM에 의해 클라이언트에 필요한 검색을 수행한다. 그리고 사용자에게 컨텍스트 변화에 따른 검색 결과를 전달한다.

2. 시스템 흐름

본 논문에서 제안하는 레벨화된 정보 검색 기법의 순차적 흐름은 그림 2에서와 같이 세 단계를 수행한다.

첫 번째 단계에서는 데이터마이닝 기법^[8] 중에서 연관 규칙 기법 (association rule mining)을 이용하여 권유 정보 규칙을 추출한다. 이를 통해 컨텍스트 데이터베이스로부터 컨텍스트 값과 관련된 권유 정보 규칙을 생성한다.

두 번째 단계에서는 현재 컨텍스트 값과 권유 정보 규칙을 사용하여 가까운 미래에 나타날 컨텍스트 값별 권유 정보를 추출한다. 이 때, 현재 컨텍스트 값을 기준으로 가까운 미래에 도달할 컨텍스트 값이란 예를 들면, 위치 컨텍스트의 경우는 현재 위치로부터 일정한 거리 이내에 있는 위치가 될 수 있다.

그리고 마지막 단계에서는 컨텍스트 값별 권유 정보와 레벨화 정책, 사용자 프로파일, 접근 점수를 이용하여 가장 가까운 미래에 나타날 레벨화된 컨텍스트 값별 권유 정보를 추출한다. 두 번째 단계를 통해 추출된 컨텍스트 값별 권유 정보는 저장 용량의 증가와 전송 속도의 증가를 초래할 수 있다는 문제점이 있다. 따라서 본 논문에서는 가장 가까운 미래에 사용될 수 있는 레벨화된 컨텍스트 값별 권유 정보만을 저장하고, 이 과정을 반복적으로 수행하는 기법을 사용한다. 이 때, 레벨화 정책은 해당 컨텍스트의 특성과 어플리케이션 상황등을 고려하여 결정된다. 레벨화 정책을 수립할 때 고려되어야 할 컨텍스트에는 계속적으로 변화하는 특성을 가진 위치, 시간, 기온 등이 있다. 본 논문에서는 이

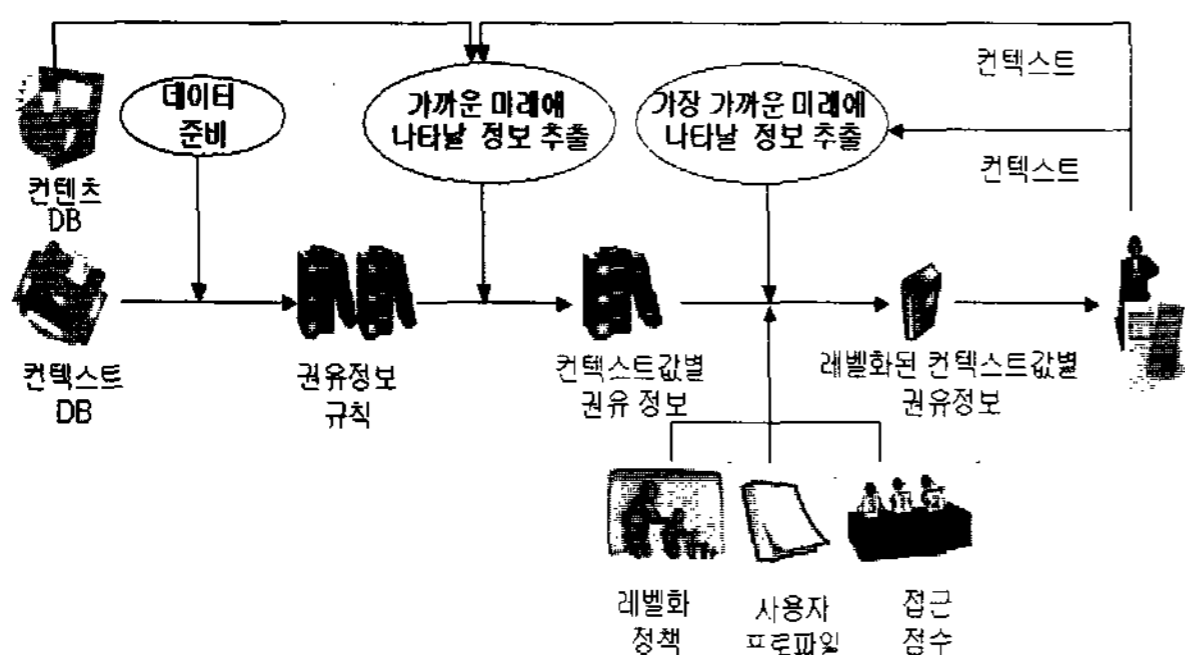


그림 2. 시스템 흐름도

Fig. 2. Flow Diagram of System.

러한 레벨화 정책은 주어진다고 가정하고, 예로서만 설명하고자 한다. 위치 컨텍스트를 예로 들면, 속도 등과 같은 특성을 이용하여 레벨을 결정할 수 있다. 즉, 속도가 빠를수록(느릴수록) 보다 넓은(좁은) 위치에 대한 간결한(자세한) 정보를 원하는 것으로 보고, 더 작은(큰) 레벨 값을 가지게 된다.

IV. 레벨화된 정보검색 알고리즘

1. 기본 모델

본 장에서는 그림 2에서 나타난 수행 단계 중 레벨화된 컨텍스트 값별 권유 정보를 검색하는 알고리즘을 제안한다. 첫 번째 단계와 두 번째 단계에 대한 알고리즘은 [7]을 참고하도록 한다.

제안하는 검색 알고리즘은 계속적으로 변화되는 컨텍스트 특성을 고려하여 다음과 같은 기법을 사용한다.

- 정보 검색이 요구될 때마다 현재의 컨텍스트 값이 어떠한 레벨 값에 해당하는지 결정한다. 이렇게 결정된 레벨 값은 검색될 데이터의 상세화 정도를 결정하는 중요한 요소로 작용한다.
- 컨텍스트 값의 계속된 변경에도 즉각적으로 검색 결과를 제공하기 위해 데이터의 지역성을 이용한다. 이를 위해 프리페칭(prefetching)과 교체(replacement) 기법을 사용한다.
- 프리페칭과 교체 대상을 선정할 때 컨텍스트의 레벨화 특성을 고려한 알고리즘을 제안한다. 이를 통해 사용자들은 현재의 컨텍스트 값에 따라 필요한 데이터만을 신속하게 검색할 수 있게 된다.

2. 프리페칭과 교체 대상 선정

프리페칭할 권유 정보를 선정하기 위해서는 미래에 접근될 정보를 올바르게 선정하는 일이 중요하다. 이를 위해 본 논문에서는 다음과 같은 2단계 절차를 따른다.

- 1단계 : 컨텍스트 값별로 결정된 레벨 값에 따라 프리페칭 후보 구간이 결정되고, 이를 통해 후보 컨텍스트 값이 선정된다. 이 때 후보 구간은 임계치(threshold) 값에 의해 설정된다.
- 2단계 : 지역 공간의 제한점을 극복하면서 미래에 접근될 데이터를 보다 적합하게 선정하기 위해서는 후보 컨텍스트 값과 관련된 모든 권유 정보를 프리페칭할 수는 없다. 따라서 본 논문에서는 접근 점수

(access score)를 제안한다. 각 권유 정보당 접근 점수가 할당되며, 이 때, 접근 점수가 크면 클수록 미래에 접근될 가능성이 높다는 것을 의미하게 된다. 따라서 이미 프리패칭되어 있는 권유 정보의 접근 점수가 후보 컨텍스트 값으로부터 추출된 권유 정보의 접근 점수보다 작은 값이라면 교체된다. 다음 식은 각 권유 정보에 대한 접근 점수를 정의한다.

$$\text{접근 점수} = a * \text{우선순위 점수} + b * \text{사용자 프로파일 점수}$$

여기서, $a+b=1$, $0 \leq \text{우선순위 점수} \leq 1$, $0 \leq \text{사용자 프로파일 점수} \leq 1$ 이다. 그리고 우선순위 점수는 계속적으로 변화하는 컨텍스트 값에 의해 결정된 레벨 값에 의해 설정되며, 사용자 프로파일 점수는 사용자가 명시적으로 입력한 컨텍스트 값별 중요도를 의미한다.

각 권유 정보에 대한 우선순위 점수는 그림 2의 첫 번째 단계를 통해 생성된 권유 정보 규칙을 기반으로 얻어진다. 이를 위해 본 논문에서는 연관 규칙의 신뢰도(confidence) 값을 레벨 값과 연관시켰다. 즉, 작은 레벨 값은 큰 신뢰도 값을 갖는 권유 정보 규칙에 해당하

표 1. 프리패칭과 교체 알고리즘
Table 1. Prefetching and Replacement Algorithm.

입력 : 컨텍스트 값, 권유 정보 규칙, 컨텍스트 값별 권유 정보 출력 : 레벨화된 컨텍스트 값별 권유 정보 방법 : 1단계) 프리패칭 IF (이전 컨텍스트 값에 따른 레벨 값 >= 현재 컨텍스트 값에 따른 레벨 값) THEN IF (현재 레벨 값에 의해 결정된 구간에 추가될 후보 컨텍스트 값이 있는가?) THEN 1. 추가될 후보 컨텍스트 값 선정 2. 추가될 후보 컨텍스트 값과 현재 레벨 값에 의해 결정된 우선순위 점수를 가지는 후보 권유 정보 추출 3. 추출된 후보 권유 정보에 대해 접근 점수 계산 ELSE 1. 현재 레벨 값에 의해 결정된 구간에 속하는 후보 컨텍스트 값과 우선순위 점수 선정 2. 후보 컨텍스트 값에 대해, 이전 레벨 값에 의한 우선순위 점수와 비교하여 더 낮은 우선순위 점수를 가지는 후보 권유 정보만을 추출 3. 추출된 후보 권유 정보에 대해 접근 점수 계산 END IF ENDIF 2단계) 교체 IF (후보 권유 정보의 접근 점수 > 이전 프리패칭 권유 정보의 접근 점수) THEN 이전 프리패칭 권유 정보 교체 END IF

는 권유 정보를 프리패칭하고, 큰 레벨 값은 작은 신뢰도 값을 갖는 권유 정보 규칙에 해당하는 권유 정보까지 모두 프리패칭하게 된다.

3. 프리패칭과 교체 전략

본 논문에서는 컨텍스트의 계속된 변경에도 즉각적으로 검색 결과를 제공하기 위해 프리패칭 기법을 사용한다. 이를 위해 미래에 접근될 권유 정보를 선택하여 지역적으로 저장한다. 그러나 프리패칭될 저장 용량에는 제한이 있기 때문에 미래에 접근되지 않을 권유 정보를 교체하는 교체 전략을 함께 사용한다. 표 1은 프리패칭과 교체 알고리즘을 보여준다.

V. 실험 평가

1. 실험 환경

실험을 위해 본 논문에서 제안하는 레벨화된 정보 검색 기법에 의한 System1과 그렇지 않은 System2를 구현하였다. System2는 System1과 비교하기 위해 가장 상세한 정보만을 사용하여 구현하였다. 자바를 사용하여 구현하였으며, 클라이언트는 펜티엄 랩탑에서 실행하고, 서버는 펜티엄 데스크 탑에서 실행하였다. 또한, 실험을 보다 분명하게 하기 위해서 네트워크 요소는 고려하지 않았으며, System2는 System1과 동일한 접근 점수 기법을 사용하였다.

본 실험에서는 다음과 같은 가상 데이터를 사용하였다. 컨텍스트 값의 개수는 100, 권유 정보의 개수는 콘텐츠 개수의 0.005, 그리고 권유 정보 규칙의 개수는 각 컨텍스트 값별로 권유 정보의 개수만큼으로 하였다. 그리고 콘텐츠와 컨텍스트 값은 1부터 각 개수까지 생성하였다. 권유 정보 규칙의 조건 부분은 생성한 모든 컨텍스트 값으로 하고, 액션 부분은 콘텐츠로부터 랜덤 추출하여 생성하였다. 각 권유 정보에 대한 접근 점수는 랜덤하게 생성하였으며, 전체 레벨의 수는 4로 하였다. 이때, 각 레벨 별로 요구되는 컨텍스트의 개수는 레벨 1 (2,3,4)에서 10 (7,4,1)으로 하였다. 그리고 컨텍스트 값이 변경될 때마다 레벨 값이 커지는 경우를 실험하였다.

2. 실험 결과

본 실험에서는 콘텐츠 개수의 변화와 클라이언트의 저장 용량 변화에 대한 영향을 비교하였다. 그리고 이를 레벨별로 비교해보았다.

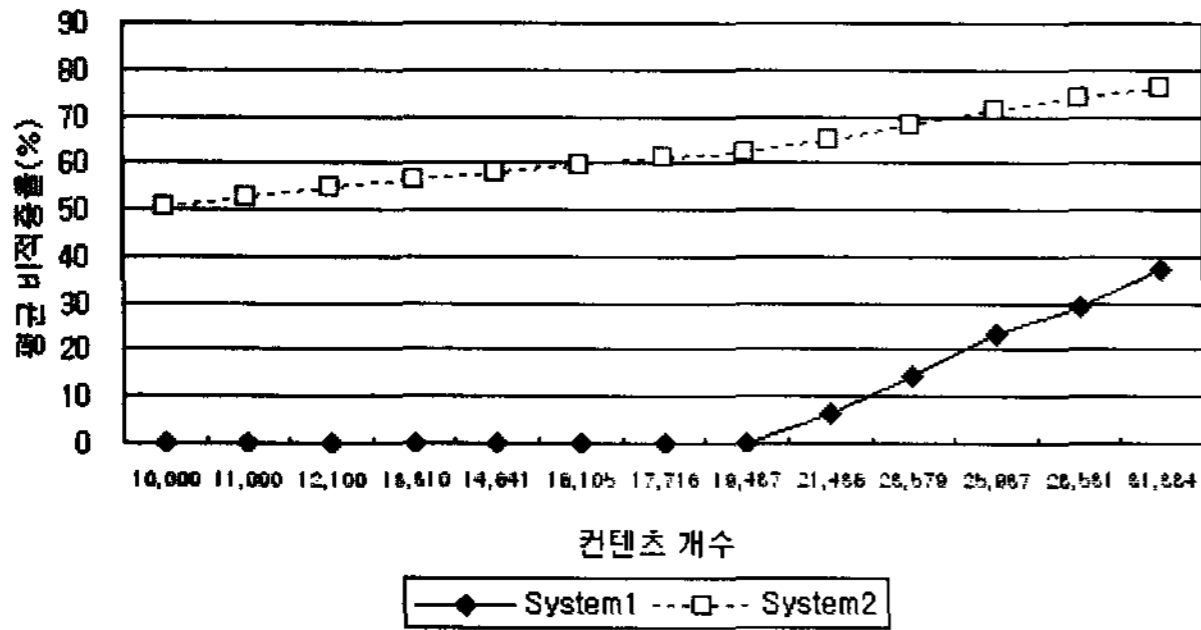


그림 3. 컨텐츠 개수 변화에 따른 비교
Fig. 3. The Impact of the Number of Contents.

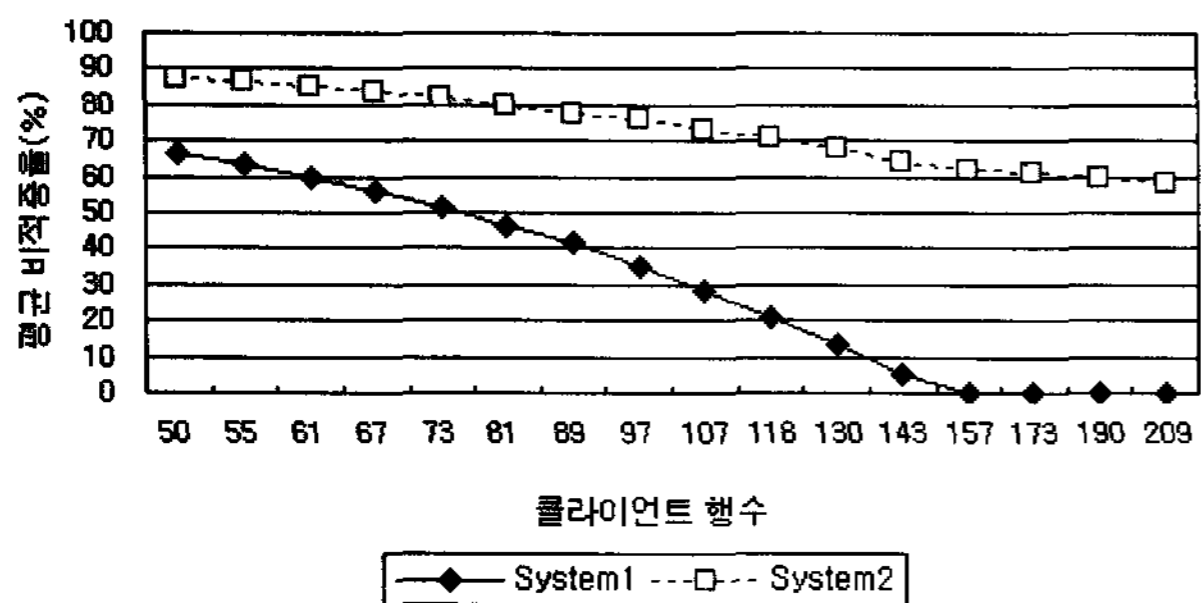


그림 4. 클라이언트 저장 용량 변화에 따른 비교
Fig. 4. The Impact of the Client's Storage Size.

컨텐츠 개수의 변화를 실험하기 위해 클라이언트의 저장 용량은 데이터 최대 수를 100으로 고정된 상태에서, 컨텐츠의 수를 10,000부터 10%씩 증가하면서 31,384 까지 그 변화를 측정하였다. 그림 3은 컨텐츠 개수의 변화에 대한 컨텍스트별 클라이언트 캐쉬 평균 비적중율(miss ratio)을 측정한 실험 결과를 보여준다.

클라이언트의 저장 용량 변화 실험을 위해 컨텐츠 개수는 50,000으로 고정하고, 클라이언트의 최대 수를 50부터 209까지 10%씩 증가하면서 그 변화를 측정하였다. 그림 4는 클라이언트의 저장 용량 변화에 대한 컨텍스트별 클라이언트 캐쉬 평균 비적중율을 측정한 실험 결과를 보여준다.

그림 3과 그림 4에서 보는 바와 같이 System1에서는 컨텐츠의 수가 감소하거나 클라이언트의 용량이 감소할 때 비적중율은 거의 0에 가까워지는 것을 볼 수 있다. 그러나 System2는 비적중율이 모두 50% 이상임을 볼 수 있다. 또한, System1의 비적중율은 System2보다 항상 낮음을 볼 수 있다. 이를 통해 제안 기법이 기존 기법에 비해 검색 성능이 향상됨을 알 수 있으며, 적절한 컨텐츠의 수와 클라이언트의 용량이 주어질 때는 검색 성능이 매우 좋아지게 됨을 알 수 있다.

또한 이러한 결과를 레벨별로 관찰할 목적으로 컨텐츠 개수의 변화와 클라이언트의 저장 용량 변화 모두에

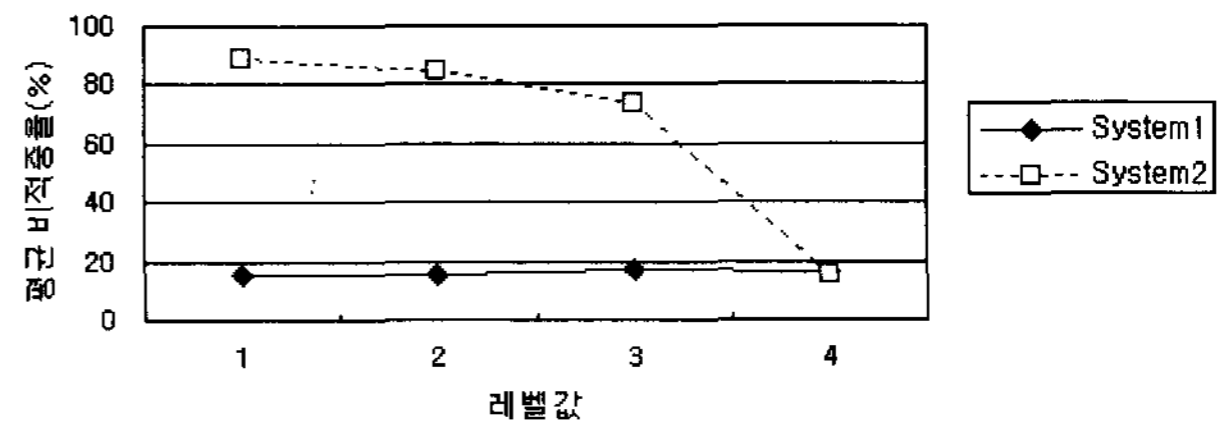


그림 5. 컨텐츠 개수 변화에 대한 레벨별 비교
Fig. 5. The Impact of the Number of Contents by Levels.

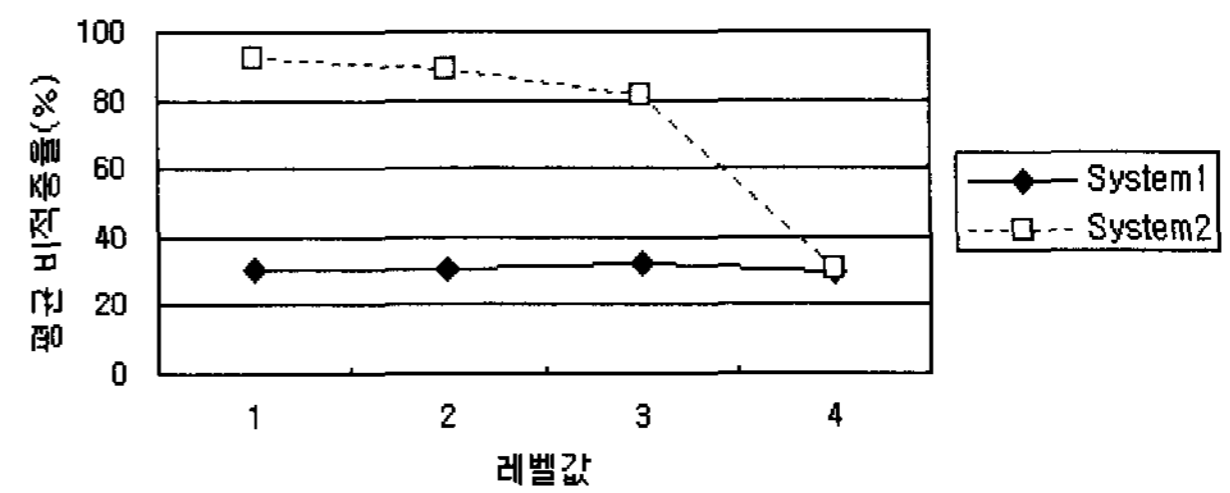


그림 6. 클라이언트 용량 변화에 대한 레벨별 비교
Fig. 6. The Impact of the Client's Storage Size by Levels.

대한 레벨별 평균 비적중율을 실험하였다.

그림 5는 컨텐츠 개수의 변화에 대한 레벨별 평균 비적중율에 대한 실험 결과를 보이며, 그림 6은 클라이언트의 저장 용량 변화에 대한 레벨별 평균 비적중율을 보인다. 그림 5와 그림 6으로부터 레벨 값 1,2,3에서는 System1의 비적중율이 System2의 비적중율보다 항상 낮으며, 레벨값 4에 한해 System1과 System2의 비적중율이 거의 유사함을 알 수 있다. 이는 레벨 4에서는 System1과 System2 모두 가장 상세화된 정보를 사용하기 때문이다. 또한, 레벨 값이 작을수록 System1과 System2간의 차이가 커짐을 알 수 있다. 이를 통해 제안 기법이 기존 기법에 비해 레벨 값이 작을수록 검색 성능이 더욱 향상될 수 있음을 알 수 있다.

VI. 결 론

본 논문에서는 컨텍스트 인식 환경에서 레벨화된 정보 검색 기법을 제안하였다. 이를 위해 시스템을 설계하고 알고리즘을 제시하였다. 제안된 기법은 컨텍스트의 레벨화 특성을 활용함을 주요 특징으로 한다. 이러한 접근 기법은 컨텍스트 인식 환경에서의 데이터 검색 기법에 대한 기존 연구에서는 볼 수 없었던 새로운 방법이다.

제안된 검색 기법의 특징은 다음과 같다. 첫째, 컨텍스트의 레벨화 특성을 이용한 점진적 검색 기법을 통해

정보를 정확하고 신속하게 정보를 검색할 수 있다. 둘째, 프리패칭 기법을 활용함으로써 가장 가까운 미래에 사용될 정보만을 미리 저장함으로써 유비쿼터스 컴퓨팅에서 많이 사용되는 모바일 장치에서의 저장 공간 제약과 데이터 검색 속도의 제약을 극복하였다. 셋째, 접근점수 개념을 제안함으로써 컨텍스트 인식 환경에 적합한 새로운 교체 기법을 새롭게 제안하였다.

참 고 문 헌

- [1] Mark Weiser, "The computer for the 21st century," *Scientific American*, Vol. 265, No. 30, pp. 94-104, 1991.
- [2] Anind K. Dey, Daniel Salber, Gregory D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," Anchor article of a special issue on Context-Aware Computing *Human-Computer Interaction Journal*, Vol. 16, No. 2-4, pp. 97-166, 2001.
- [3] Paul Prekop, Mark Burnett, "Activities, Context and Ubiquitous Computing," Special Issue on Ubiquitous Computing, *Computer Communications*, Autumn, 2002.
- [4] P. J. Brown, G. J. F. Jones, "Context-aware Retrieval: Exploring a New Environment for Information Retrieval and Information Filtering," *Personal and Ubiquitous Computing*, 2001, Volume 5, Issue 4, December, pp. 253-263, 2001.
- [5] Getri Kappel, Birgit. Proll, Werner Retschitzegger, Wieland Schwinger, "Customisation for Ubiquitous Web Applications-A Comparison of Approaches," *International Journal of Web Engineering and Technology*, Vol. 1, No. 1, pp. 79-111, January, 2003.
- [6] Bradley J. Rhodes, Pattie Maes, "Just-in-time information retrieval agents," *IBM Systems Journal*, Vol. 39, No. 3&4, pp. 685-704, 2000.
- [7] Joonhee Kwon, Sungrim Kim, Yongik Yoon, "Just-In-Time Recommendation using Multi-Agents for Context-Awareness in Ubiquitous Computing Environment," *Lecture Notes in Computer Science* 2973, Mar, 2004.
- [8] R. Agrawal, T. Imielinski, A. Swami, "Mining association rules in large databases," In *Proceedings of ACM SIGMOD Conference on Management of Data*, Washington D.C., pp. 207-216, May 1993.

저 자 소 개



김 성 림(정회원)
1994년 숙명여자대학교 전산학과
학사 졸업.
1997년 숙명여자대학교 대학원
전산학과 석사 졸업.
2002년 숙명여자대학교 대학원
컴퓨터학과 박사 졸업.

2004년 3월~현재 서일대학 인터넷정보전공
교수.

<주관심분야 : 유비쿼터스 컴퓨팅, 웹 데이터베이스, XML, 멀티미디어 의료 데이터베이스>



권 준 희(정회원)
1992년 숙명여자대학교 전산학과
학사 졸업.
1994년 숙명여자대학교 대학원
전산학과 석사 졸업.
2002년 숙명여자대학교 대학원
컴퓨터학과 박사 졸업.

2003년 3월~현재 경기대학교 정보과학부
전자계산학 전공 교수.

<주 관심분야 : 유비쿼터스 컴퓨팅 및 LBS, 공간
데이터베이스 및 GIS, 워크플로우>