

선인출 기반의 모바일 사전

(A Mobile Dictionary based on a Prefetching Method)

홍순정[†] 문양세^{**} 김혜숙^{***} 김진호^{**} 정영준^{****}
 (Soon-Jung Hong) (Yang-Sae Moon) (Hea-Suk Kim) (Jin-Ho Kim) (Young-Jun Chung)

요약 모바일 인터넷 환경에서는 학습 내용을 검색 및 다운로드하기 위하여 모바일 기기와 서버 사이에 잦은 통신이 필요하다. 본 논문에서는 모바일 사전을 사용함에 있어서, 네트워크 비용을 절감하고 통신 효율을 높이기 위한 효율적인 선인출 기법을 제안한다. 이를 위해 본 논문에서는 다음 방법으로 연구를 전개한다. 첫째, 모바일 사전을 위한 선인출 기반의 동작 프레임워크를 제안한다. 둘째, 패킷 요금 방식과 정액 요금 방식으로 구분하여 선인출할 데이터의 양을 결정하는 방법을 제시한다. 셋째, 중·고등학생 대상의 모바일 영한사전에 초점을 맞춘 선인출 데이터 결정 방법을 제안한다. 넷째, 이러한 데이터 양 및 데이터 종류 결정 방법을 바탕으로 선인출 알고리즘을 제안한다. 다섯째, 실험을 통하여 제안한 선인출 방법의 우수성을 입증한다. 이 같은 연구에 따른 본 논문의 공헌은 다음과 같이 요약할 수 있다. 첫째, 선인출을 모바일 응용에 적용한 첫 번째 시도로서 의미를 갖는다. 둘째, 선인출을 모바일 사전에 적용하기 위한 체계적인 방법론을 제시하였다. 셋째, 선인출 적용을 통해 네트워크 기반 모바일 사전의 성능 향상을 도모하였다. 실제 실험 결과, 제안한 방법은 기존의 요구인출에 평균 9.8%~33.2%의 높은 성능 향상을 나타냈다. 이 같은 결과를 볼 때, 본 논문의 연구 결과는 모바일 사전뿐 아니라 선인출 기능을 필요로 하는 무선 인터넷 기반의 여러 응용에 적용될 수 있는 우수한 연구 결과라 사료된다.

키워드 : 모바일 사전, 선인출, 무선 인터넷, 라운드트립, 모바일 응용

Abstract In the mobile Internet environment, frequent communications between a mobile device and a content server are required for searching or downloading learning materials. In this paper, we propose an efficient prefetching technique to reduce the network cost and to improve the communication efficiency in the mobile dictionary. Our prefetching-based approach can be explained as follows. First, we propose an overall framework for the prefetching-based mobile dictionary. Second, we present a systematic way of determining the amount of prefetching data for each of packet-based and flat-rate billing cases. Third, by focusing on the English-Korean mobile dictionary for middle or high school students, we propose an intuitive method of determining the words to be prefetched in advance. Fourth, based on these determination methods, we propose an efficient prefetching algorithm. Fifth, through experiments, we show the superiority of our prefetching-based method. From this approach, we can summarize major contributions as follows. First, to our best knowledge, this is the first attempt to exploit prefetching techniques in mobile applications. Second, we propose a systematic way of applying prefetching techniques to a mobile dictionary. Third, using prefetching techniques we improve the overall performance of a network-based mobile dictionary. Experimental results show that, compared with the traditional on-demand approach, our prefetching-

· 이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국 학술진흥재단의 지원을 받아 연구되었음(KRF-2007-331-D00381) 정 회 원 : 강원대학교 컴퓨터과학전공 교수 ychung@kangwon.ac.kr
 · 이 논문은 2007 한국컴퓨터종합학술대회에서 '선인출 기반의 모바일 사전'의 논문집수 : 2007년 9월 27일 제목으로 발표된 논문을 확장한 것임 심사완료 : 2008년 2월 12일

† 학생회원 : 강원대학교 컴퓨터교육과 hayaharu@hanmail.net
 ** 종신회원 : 강원대학교 컴퓨터과학전공 교수 ysmoon@kangwon.ac.kr (Corresponding author) jhkim@kangwon.ac.kr
 *** 학생회원 : 강원대학교 컴퓨터과학전공 hskim@kangwon.ac.kr

Copyright © 2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
 정보과학회논문지: 소프트웨어 및 응용 제35권 제3호(2008.3)

based approach improves the average performance by 9.8%~33.2%. These results indicate that our framework can be widely used not only in the mobile dictionary but also in other mobile Internet applications that require the prefetching technique.

Key words : Mobile dictionary, Prefetching, Mobile Internet, Roundtrip, Mobile applications

1. 서 론

최근 모바일 시장에서는 이동통신 사업자가 웹 사이트 업체와 제휴하여 다양한 형태의 모바일 사전을 제공하고 있다[1-3]. 이와 같은 모바일 사전은 1) 서버와 접속하여 프로그램을 다운로드 받은 후 모바일 기기 상에서 독자적으로 구동되는 단독(stand-alone)형과 2) 네트워크를 이용하여 지속적으로 서버와 통신하는 네트워크(network-based)형의 두 가지 형태로 나뉘어 서비스되고 있다. 모바일 사전의 경우, 단독형보다는 네트워크형이 더 많은 단어를 제공할 수 있어 폭넓은 이용이 가능하고 수정이 용이한 특징을 가진다. 그러므로, 본 논문에서는 이러한 네트워크형 모바일 사전의 단점인 추가적인 네트워크 비용을 최소화하기 위하여 선인출 기법을 활용한다.

본 논문에서는 클라이언트-서버 모델에서의 선인출 기법을 모바일 사전에 적용한다. 일반적인 클라이언트-서버 모델에서, 서버는 데이터를 저장하고 관리하는 역할을 하며, 클라이언트는 서버에 데이터 인출을 요구하거나 인출된 데이터를 관리하는 역할을 한다[4]. 이 때, 서버의 데이터들을 클라이언트 메모리로 인출하는 방식은 1) 데이터가 필요할 때마다 서버에서 인출하는 요구인출(on-demand fetch)과 2) 향후 액세스될 데이터들을 서버에서 미리 인출하는 선인출(prefetch)로 분류할 수 있다. 선인출 방법은 미리 인출된 데이터들이 추후 서버와의 라운드트립(roundtrip) 횟수가 줄고 성능이 향상되는 장점이 있다[4-7]. 본 논문에서는 이러한 클라이언트-서버 모델의 선인출 개념을 모바일 사전에 적용하여 네트워크 비용을 줄이고 성능을 향상시킨다.

모바일 사전을 위한 선인출 기법은 다음과 같이 동작한다. 우선, 사용자가 검색을 원하는 단어 w_1 을 모바일에서 서버로 요청하면, 서버에서는 요청한 단어 w_1 에 해당하는 결과 r_1 뿐 아니라, 다음 검색 시에 요청 가능성이 높은 단어 w_2 와 w_3 을 예측하여 이들의 결과인 r_2 와 r_3 을 한꺼번에 모바일에 전송해준다. 전송된 결과인 r_2 와 r_3 는 모바일의 메모리에 캐싱(caching)되어 관리된다. 그런 다음, 사용자가 다음 단어로 w_2 나 w_3 을 요청하는 경우에는 서버에 접속할 필요 없이 모바일에 이미 선인출된 r_2 나 r_3 을 결과로서 제공할 수 있다. 이와 같이, 모바일 사전에 선인출 기법을 적용하면, 서버로의 라운드트립 횟수를 줄일 수 있을 뿐 아니라, 패킷 사용 효율

개선과 함께 전체 성능을 향상시킬 수 있다.

그런데 이와 같은 선인출 기법을 사용하여 서버에서 한 번에 가져올 수 있는 데이터 양은 클라이언트 메모리 양과 모바일 네트워크의 통신비용에 의존적이다. 따라서 본 논문에서는 패킷 요금 방식과 정액 요금 방식에 따라 선인출할 데이터 양을 결정한다. 패킷 요금 방식에서는 과금이 패킷 단위로 이루어진다. 이러한 패킷 요금 방식의 경우 데이터 양에 따른 비용 상승의 문제가 발생할 수 있으므로, 선인출하는 데이터 양은 전송되는 패킷 내에서 원하는 데이터 이외의 여분 공간으로 한정한다. 정액 요금 방식에서는 전송하는 데이터 양에 상관없이 매달 일정 요금을 지불하고 무제한으로 데이터를 이용한다. 이 방식의 경우 선인출로 인한 추가의 네트워크 비용이 발생하지 않으므로 선인출하는 데이터 양은 모바일 기기의 메모리 양에 제한된다. 본 논문에서는 이와 같이 과금 방식에 따라 선인출 데이터 양을 다르게 계산하는 선인출 방법을 제안한다.

다음으로, 선인출할 데이터 내용은 응용이나 사용자 종류에 따라 달라질 수 있다. 본 논문에서는 중·고등학교 생이 사용하는 모바일 영한 사전에 초점을 맞추어 선인출할 데이터를 결정한다. 본 논문의 방법은 중·고등학교 교과서의 각 과별로 문단 아래 제시되는 새로운 단어를 한꺼번에 가져오는 간단한 선인출 방식이다. 그 이유는 교과서의 새로 나오는 주요 단어들은 일반적으로 학생들이 잘 모르는 경우가 많기 때문이다. 즉, 교과서의 새로운 단어들은 짧은 시간 내에 함께 혹은 연속적으로 액세스될 가능성이 높아 선인출 효과를 극대화할 수 있기 때문이다. 이에 따라, 본 논문에서는 중·고등학교 대상의 모바일 영한사전에 초점을 맞추어, 교과서의 각 과별로 문단 아래 제시되는 새로운 단어를 한꺼번에 갖고 오는 선인출 알고리즘을 제시한다.

제한한 선인출 기법이 모바일 사전의 성능 향상에 유용함을 보이기 위해, 본 논문에서는 설문조사를 통해 현실의 구체적인 데이터를 수집한 후 실제 분석을 수행하였다. 실험에서는 패킷 요금과 정액 요금 방식에 따라 선인출 단어의 개수에 따른 선인출 비율과 라운드트립 횟수를 계산하였다. 실험 결과, 제안한 선인출 방법을 모바일 영한사전에 사용할 경우, 한 개에서 수 개의 단어에 대한 선인출만으로도 높은 선인출 효과를 얻을 수 있고, 이를 통해 라운드트립 횟수를 크게 줄일 수 있는 것으로 나타났다. 결국, 제안한 선인출 방법은 모바일

사전의 네트워크 비용을 줄이고 성능을 향상시킬 수 있는 매우 효율적 방법이라 사료된다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련연구로 기존의 모바일 관련 연구와 선인출 기법을 소개한다. 제3장에서는 모바일 사전에 선인출 기법을 적용하기 위한 동작 프레임워크를 제안한다. 또한, 이를 모바일 사전에 적용하기 위하여 선인출할 데이터 양과 선인출할 데이터를 선택하는 방법을 제안한다. 제4장에서는 설문조사 결과를 기반으로 선인출 비율 및 라운드트립 횟수를 분석한 결과를 제시하고, 그 의미를 해석한다. 마지막으로, 제5장에서 결론을 맺는다.

2. 관련 연구

무선 인터넷의 발달로 인하여 유선의 콘텐츠 및 서비스를 모바일로 옮기고자 하는 많은 노력이 있었다. 예를 들어, 참고문헌 [8]에서는 동적인 웹 콘텐츠를 모바일 환경에서 사용하기 위한 적응형 기법을 연구하였다. 또한, 참고문헌 [9]에서는 유선의 웹 콘텐츠를 모바일 환경에서 효율적으로 사용하기 위한 모바일 사용자 에이전트 개념을 제시하였다. 그리고, 참고문헌 [10]에서는 멀티미디어 콘텐츠를 작은 화면의 제약이 있는 모바일 기기로 옮기기 위한 포맷 변환 방법에 대해 연구하였다. 이와 같이 기존 연구들이 콘텐츠 및 서비스를 유선 환경에서 모바일 환경으로 옮기는데 연구의 초점이 맞추어져 있었다. 반면에, 본 연구에서는 콘텐츠를 검색 및 다운로드하기 위하여 모바일 기기에서 서버로 접속할 때 네트워크 비용을 절감하고 통신 효율을 높이기 위한 선인출 기법을 연구한다.

최근 모바일 시장에서는 이동통신 사업자가 웹 콘텐츠 업체와 제휴하여 다양한 모바일 사전 서비스를 제공하고 있다. 대부분의 모바일 소프트웨어가 그러하듯이 모바일 사전의 동작 방식 역시 크게 단독형과 네트워크형이 있다. 단독형은 애플리케이션 및 콘텐츠 사업자가 제공하는 프로그램을 다운로드 한 후, 서비스 이용 시에는 별도의 네트워크 접속 없이 오프라인 상에서 구동하는 방식이다. 반면에, 네트워크형은 프로그램을 다운로드한 후에도, 서비스 이용을 위해서 네트워크를 지속적으로 이용하는 구동 방식을 말한다. 단독형은 일단 다운로드 받은 이후에는 추가의 네트워크 비용이 발생하지 않고 검색속도가 빠른 장점이 있는데 반하여, 메모리가 많이 필요하고 메모리 용량의 한계로 인하여 소규모 사전에만 적합하다는 단점이 있다. 단독형은 또한 사전의 내용이 수정될 경우 각 모바일 기기의 내용을 수정해야 하는 번거로움이 따른다. 이에 반하여, 네트워크형은 서비스 이용 시마다 비용이 발생하기는 하지만, 단독형보다는 더 많은 단어를 사전에 수록할 수 있는 장점이 있

다. 또한, 사전의 내용이 수정되어도 모바일 기기의 프로그램은 수정할 필요가 없다는 장점이 있다. 이와 같이 모바일 사전을 사용할 경우, 단독형보다는 네트워크형이 보다 많은 단어를 제공할 수 있으며, 유연한 구조를 제공한다. 본 논문에서는 이러한 네트워크형 모바일 사전의 단점인 추가적인 네트워크 비용을 최소화하기 위하여 클라이언트-서버 모델에서 제안한 선인출 기법을 활용한다.

클라이언트-서버 모델에서 서버의 데이터들을 클라이언트 메모리로 인출하는 방식은 크게 요구인출과 선인출로 분류할 수 있다[4,5]. 요구인출이란 선인출이든 일단 클라이언트에 전달된 데이터는 클라이언트 메모리에 캐싱되어 추후 동일한 요구가 있을 경우에 재사용될 수 있다. 실제로 이와 같은 캐싱을 통해 성능이 향상되기도 하나, 본 논문에서는 선인출 효과에만 초점을 맞추기 때문에 캐싱을 통한 재사용은 분석에서 제외하였다. 먼저, 요구인출은 클라이언트에서 데이터가 필요할 때마다 매번 서버에서 인출하는 방법이다. 이 요구인출 방법은 꼭 필요한 데이터만 인출한다는 장점이 있으나, 데이터가 인출될 때마다 서버와의 라운드트립이 발생하므로, 성능이 저하되는 단점이 있다. 이에 반해, 선인출은 향후 액세스될 데이터들을 서버에서 미리 인출하는 방법이다. 선인출 방법은 미리 인출된 데이터들이 향후 실제 사용되는 경우 서버와의 라운드트립 횟수가 줄고 성능이 향상되는 장점이 있다. 그러나 선인출된 데이터들이 응용에서 실제로 액세스되지 않으면 불필요한 데이터들의 선인출 비용으로 인하여 시스템의 성능이 오히려 저하될 수 있다[4].

이러한 선인출 기법은 전통적인 클라이언트-서버 모델, 객체 관계형 모델 등에서 시스템의 전체 성능을 향상시키기 위해 많이 사용되었다. 클라이언트-서버 모델에서는 서버와 클라이언트 사이의 라운드트립의 횟수를 줄이고, 성능을 향상시키기 위한 목적으로 사용되었다[11,12]. 객체 관계형 모델의 경우도 객체관계형 DBMS 등에서 라운드트립 횟수를 줄이기 위해 다양한 선인출 기법이 연구되었다[4,5]. 그리고, 클러스터 웹 서버에서도 서비스 지연을 감소시키기 위한 서버 노드 사이의 자료 선인출 기법을 연구하였다[13,14]. 본 논문에서는 이 중 가장 일반적인 클라이언트-서버 모델에서 사용한 선인출 개념을 모바일 사전에 적용하는 방법을 사용한다.

기존의 선인출 기법은 선인출 대상을 선정하는 방식에 따라 크게 페이지 기반 선인출, 액세스 패턴 기반 선인출, 사용자 힌트 기반 선인출, 그리고 문맥 기반 선인출의 네 가지로 구분할 수 있다[4]. 첫째, 페이지 기반 선인출은 요청된 객체(혹은 레코드)를 서버에서 인출할 때, 그 객체가 속한 페이지에 있는 다른 객체들을 함께

인출하는 방법이다[15]. 이 방법은 페이지 내의 객체들이 연속되어 액세스되는 응용에서 좋은 선인출 효과를 나타낸다. 둘째, 액세스 패턴 기반 선인출은 최근의 객체 참조에 기반하여 향후의 액세스 패턴을 예측하는 방법이다[16]. 이 방법은 응용이 주로 사용하는 액세스 패턴이 일정하거나 주기적인 경우에 선인출 효과가 크게 나타난다. 셋째, 사용자 힌트 기반 선인출은 사용자가 제시한 힌트에 기반하여 선인출할 대상을 결정하는 방법이다[17]. 이 방법은 사용자가 선인출에 간접적으로 관여하는 방법으로, 사용자가 얼마나 유용한 힌트를 제공하느냐에 따라 선인출 효과가 결정된다. 넷째, 문맥 기반 선인출은 객체를 인출할 때, 그 객체의 구조 문맥(structure context)이 가리키는 구조의 모든 객체들도 함께 인출하는 방법이다[18]. 이 방법은 객체들의 구조 문맥이 잘 정의되어 있고, 또한 응용이 정의된 구조 문맥에 따라서 객체들을 액세스할 때 우수한 성능을 나타낸다.

본 논문에서는 “중고등학교 교과서 기반의 선인출”을 수행하는데, 이는 사용자 힌트 기반의 선인출로 볼 수 있다. 즉, 중고등학교 학생인 사용자가 요청한 단어를 힌트로 하여 선인출 대상을 결정하므로, 사용자 힌트 기반 선인출의 특수한 예라 생각할 수 있다. 이와 같이 본 논문에서 사용한 사용자 힌트 기반 선인출 이외에 다른 선인출 기법을 모바일 사전에도 적용할 수 있다. 그러나, 이들 나머지 방법들은 모바일 환경에는 구현이 복잡하며, 선인출 효과를 직관적으로 장담할 수도 없다. 따라서, 본 논문에서는 선인출의 다양한 방법들을 다루기 보다는, 선인출 자체에 초점을 맞추어 선인출을 사용한 경우와 그렇지 않은 경우만을 제4장의 성능 평가에서 비교한다. 그리고, 사용자 힌트 기반 선인출 방법을 제외한 다른 선인출 방법과의 비교 분석은 향후 연구로 남겨둔다.

3. 선인출 기반 모바일 사전

3.1 전체 프레임워크

선인출의 효과를 설명하기 위하여, 모바일 사전에서 선인출을 사용하지 않는 경우(요구인출을 사용하는 경우)와 선인출을 사용하는 경우의 동작 절차를 소개한다. 먼저, 그림 1은 선인출을 사용하지 않는 경우의 동작 절차를 나타낸다. 그림에서와 같이, 사용자가 검색을 원하는 단어 w_1 을 모바일에 입력하면, 서버에서는 그에 해당하는 결과 r_1 을 모바일에 전송해 준다. 또한, 단어 w_2 를 요청하면 같은 방식으로 w_2 에 해당하는 r_2 가 인출되어 모바일에 전송된다. 이와 같이, 선인출을 사용하지 않을 경우에는 필요할 때마다 서버에 요청을 해야 하고, 이 과정에서 빈번한 라운드트립이 발생한다.

다음으로, 그림 2는 선인출을 사용한 경우의 동작 절차를 나타낸다. 그림을 보면, 단어 w_1 을 모바일에 입력하면, 서버에서는 요청한 단어 w_1 에 해당하는 결과 r_1 뿐 아니라, 다음 검색 시 요청 가능성이 높은 단어 w_2 와 w_3 을 예측하여 이들의 결과인 r_2 와 r_3 을 한꺼번에 모바일에 전송해 주고, 모바일에서는 이들 r_2 와 r_3 을 메모리에 캐싱해 둔다. 그런 다음, 사용자가 다음 단어로 w_2 나 w_3 을 요청하는 경우에는 서버에 접속할 필요 없이 모바일에 이미 선인출되어 캐싱되어 있는 r_2 나 r_3 을 결과로서 제공할 수 있다. 이와 같이, 모바일 사전에 선인출 기법을 적용하면, 라운드트립의 횟수를 줄일 수 있을 뿐만 아니라, 이를 통해 네트워크 비용을 줄일 수 있다.

그런데 그림 2와 같은 선인출 기법을 사용하는데 있어서 서버로부터 한 번에 가져와 모바일의 메모리에 캐싱할 수 있는 데이터 양은 제한적이다. 따라서, 선인출 기법을 모바일 사전에 적용하기 위해서는 1) 선인출할 데이터의 양과 2) 선인출할 데이터의 내용을 결정해야 한다. 이에 따라, 다음 제3.2절에서는 모바일 인터넷 환

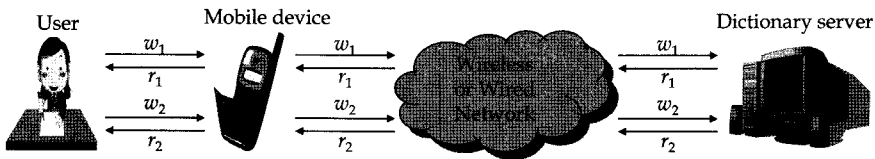


그림 1 선인출을 사용하지 않는 경우(요구인출을 사용하는 경우)의 모바일 사전 동작 절차

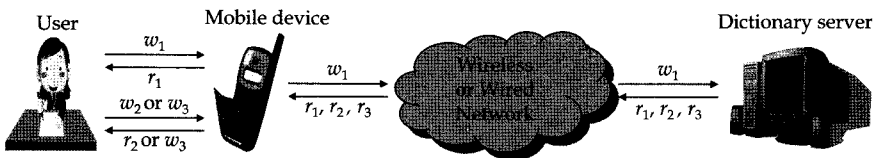


그림 2 선인출을 사용한 경우의 모바일 사전 동작 절차

경을 고려하여 선인출할 데이터 양을 결정하는 방법을, 제3.3절에서는 영한사전이라는 응용에 초점을 맞추어 선인출할 데이터 내용을 선택하는 방법을 제안한다.

3.2 선인출 데이터 양

일반적으로 선인출할 데이터의 양은 클라이언트의 메모리 양에 제한적이다[4]. 그러나 모바일의 경우 이러한 메모리 양 뿐 아니라 모바일 네트워크의 통신비용도 고려해야 한다. 최근의 모바일 네트워크는 이동통신 등의 패킷 요금 방식과 휴대 인터넷 등의 정액 요금 방식으로 구분할 수 있다. 따라서, 본 논문에서는 사용하는 요금 방식에 따라 다음 같이 선인출할 데이터 양을 결정한다.

1) 패킷 요금 방식: 전송한 데이터 양을 기준으로 과금하는 방식으로 주로 패킷 단위로 과금이 이루어진다. 이 방식은 이동통신 분야에서 주로 적용되는 요금 방식이다. 패킷 요금 방식에서는 선인출할 데이터 양에 제한을 두지 않을 경우, 패킷 양의 증가에 따른 비용 상승의 문제가 발생할 수 있다. 그러므로 선출되는 데이터의 양은 전송되는 패킷 내에서 원하는 데이터 이외의 여분 공간으로 한정한다. 예를 들어, 패킷 크기가 512 바이트이고, 요청한 단어의 양이 150 바이트라면 여분의 362 바이트 내에서 선인출할 데이터를 전송하는 것이다.

2) 정액 요금 방식: 전송한 데이터 양에 상관없이 매달 일정 요금을 지불하고 무제한으로 데이터를 이용하는 방식이다. 이 방식은 WLAN(Wireless LAN)이나 휴대 인터넷(Wibro: Wireess Broadband Internet) 등에 적용되는 요금 방식이다. 이 방식의 경우 많은 양의 데이터를 선인출 한다고 해서 추가의 네트워크 비용이 발생하지 않는다. 따라서, 이 방식의 경우 선인출하는 데이터 양은 모바일의 메모리 양에 캐싱 가능한 범위 내에서 인출하도록 한다.

3.3 선인출 데이터 내용

선인출 데이터 내용을 결정하는 것은 응용이나 사용자 종류에 따라 달라질 수 있다. 우선 응용의 경우, 본 논문에서는 영한사전, 즉 모바일 영한사전에 초점을 맞추어 선인출할 데이터의 결정 방법을 제시한다. 다음으로 사용자의 경우, 본 논문에서는 중·고등학생에 초점을 맞추어 데이터를 결정한다. 즉, 중·고등학생 중심의 모바일 영한사전에 초점을 맞추어 선인출할 데이터를 결정한다. 현재 중·고등학교의 영어 단어는 “기본 어휘표”를 통하여 2000여개가 제시되고 있으며, 본 논문에서는 이들 영어 단어를 선인출 대상으로 삼는다. 그런데, 선인출 효과를 극대화하기 위해서는 이들 단어 중 함께 혹은 연속적으로 액세스될 단어들을 파악하는 것이 중요한 요소이다.

본 논문에서는 중·고등학교 교과서의 각 과별로 문

단 아래 제시되는 새로운 단어를 한꺼번에 가져오는 매우 간단한 선인출 방식을 이용한다. 그 이유는 교육과정에서 제시한 기본 어휘표를 중심으로 교과서가 구성되므로, 교과서의 새로 나오는 주요 어휘들은 일반적으로 학생들이 잘 모르는 경우가 많기 때문이다. 즉, 새로운 단어들은 짧은 시간 내에 함께 혹은 연속적으로 액세스될 가능성이 매우 높기 때문이다. 이에 따라, 중·고등학생에 초점을 맞추어 교과서의 각 과별로 문단 아래 제시되는 새로운 단어를 한꺼번에 갖고 오는 선인출 알고리즘을 제3.4절에서 제시하고, 제4장의 실험에서 이를 이용한다.

3.4 선인출 알고리즘

본 절에서는 선인출 알고리즘을 제시한다. 선인출 알고리즘은 모바일에서 영어 단어를 요청할 경우 서버에서는 해당 단어의 결과 값과 함께 선인출한 데이터를 모바일에 전송해주는 기능을 수행한다. 그림 3은 선인출 알고리즘을 나타낸다. 우선, 단계 (1)에서는 모바일에서 요청한 단어 w 에 해당되는 결과 값을 서버에서 검색한다. 그리고, 그 내용을 현재까지의 결과인 배열 r 에 저장한다. 단계 (3)에서는 현재까지의 결과 $r[1..i]$ 를 제외한 선인출할 수 있는 공간을 s 로 계산한다. 다음으로, 단계 (4)~(9)는 요청한 단어 w 의 결과와 함께 전송할 선인출 데이터를 선정하는 과정으로, 이 과정은 데이터 공간인 s 가 허용되는 범위만큼 반복된다. 먼저, 단계 (5)에서는 이미 인출한 단어들을 기반으로, 다음에 선인출될 단어를 선택하는 함수인 ChooseNextWord()를 호출한다. 그리고, 그 결과인 선인출할 단어를 w' 이라 한다. 그리고 단계 (7)에서는 w' 에 해당하는 내용을 검색하여, 결과인 배열 r 에 추가한다. 또한, 단계 (8)에서는 전송될 데이터의 여유 공간이 남아 있는지를 계산한다. 이와 같은 과정을 반복 수행하여 데이터의 공간이 없을 경우 검색을 종료한다. 마지막으로 단계 (10)에서는 지금까지

Algorithm PrfetchWords(word w)

- (1) Retrieve the result of w and store it into $r[1]$;
- (2) $i := 1$;
- (3) $s := \text{ComputePrefetchSpace}(r, i)$;
- (4) **while** s is enough **do**
- (5) $w' := \text{ChooseNextWord}(r, i)$;
- (6) $i := i + 1$;
- (7) Retrieve the result of w' and store it into $r[i]$;
- (8) $s = \text{ComputePrefetchSpace}(r, i)$;
- (9) **end-while**
- (10) **return** $r[1..i]$ as the result;

그림 3 모바일 사전을 위한 선인출 알고리즘

Function ComputePrefetchSpace(result $r[]$, number n)

- (1) $CurrentSpace := 0;$
- (2) **for** $i := 1$ to n **do**
- (3) $CurrentSpace := CurrentSpace + \text{sizeof}(r[i]);$
- (4) $spare := MAX_SIZE - CurrentSpace;$
- (5) **return** $spare;$

그림 4 선인출을 위한 여유 공간 계산 함수

구성한 결과인 배열 $r[1..i]$ 를 클라이언트(모바일)에 전송한다. 이러한 과정을 통해 전송된 데이터는 클라이언트의 메모리에 캐싱되어 관리되며, 추후 캐싱된 데이터가 요청될 경우 서버와의 통신 없이 바로 사용될 수 있다.

그림 4는 현재까지 (선)인출한 결과인 배열 r 을 변수로 받아 추가로 선인출할 수 있는 공간을 계산하는 함수를 나타낸다. 먼저, 단계 (1)에서는 초기 사용 공간을 나타내는 $CurrentSpace$ 의 값을 0이라 설정한다. 그리고, 단계 (2)와 (3)은 현재까지의 결과인 r 의 사용 공간을 계산하는 과정이다. 단계 (3)에서는 현재까지의 사용 공간 $CurrentSpace$ 에 i -번째 단어의 결과인 $r[i]$ 의 데이터 크기를 계산하여 증가시킨다. 다음으로, 단계 (4)에서는 데이터의 허용 범위인 MAX_SIZE 에서 현재까지의 사용 공간 $CurrentSpace$ 를 빼서 여유 공간 $spare$ 를 계산한다. 허용 범위 MAX_SIZE 는 제3.2절에서 설명한 바와 같이 패킷 요금 방식인지 정액 요금 방식인지에 따라 다르게 설정되며, 단말기 종류에 따라 달라질 수 있다. 마지막으로, 지금까지의 결과인 선인출 데이터를 저장할 수 있는 여유 공간 $spare$ 를 반환한다.

그림 5는 현재까지 (선)인출한 결과인 배열 r 을 토대로 다음에 선인출할 단어를 선택하는 함수를 나타낸다. 본 논문에서는 중·고등학교 교과서의 새로운 단어를 대상으로 다음 선인출 단어를 결정한다. 따라서, 최근에 (선)인출한 단어 다음에 나오는 새로운 단어를 선인출 대상으로 삼는다. 우선, 단계 (1)에서는 교과서의 각 과별 새로 나온 단어가 저장된 목록에서 가장 최근의 결과 값 $r[n]$ 다음에 나오는 새로운 단어를 선인출 대상으로 선정하여 w 로 지정한다. 그리고, 단계 (2)에서는 결과인 w 를 반환한다. 이 때, 가장 최근의 단어 $r[n]$ (정확히는 첫 번째 시도인 $r[1]$)이 새로운 단어 목록에 없는 경우가 있을 수 있다. 이 경우는 선인출을 수행하지 않는다. 그러나, 본 논문에서는 알고리즘을 간단히 하기

Function ChooseNextWord(result $r[]$, number n)

- (1) $w :=$ the next word of $r[n]$ in the list of new words;
- (2) **return** $w;$

그림 5 선인출 단어 선택 함수

위하여 이러한 경우를 고려하지 않고 알고리즘을 기술하였다.

4. 성능 평가

4.1 실험 데이터 및 환경

중·고등학생 대상의 영한사전에 대한 선인출 적용의 효과를 분석하기 위해 설문조사를 통해 현실의 구체적인 데이터를 수집하였다. 설문조사는 강원도 춘천지역 중학교 두 곳의 2학년 다섯 학급을 대상으로 하였다. 설문 내용은 각 학교 해당 학년의 교과서를 토대로 아직 배우지 않은 단원 중 세 단원을 선택하여 교과서 지문 일부를 발췌하여 사용하였다. 학생들이 검색하는 단어를 알아보기 위함이므로, 교과서 지문을 읽으며 생소한 단어나 의미를 잘 모르는 등 영한사전 이용이 필요한 단어에 동그라미를 치게 하였다. 데이터 수집을 위한 설문조사 기간은 2006년 6월 1일부터 2006년 6월 15일까지였으며, 설문에는 총 151명(A 중학교: 58명, B 중학교: 93명)이 참여하였다. 응답한 내용 중 지문의 일부 단원에만 동그라미를 치거나 지문의 모든 단어에 동그라미를 치는 등의 불완전 응답 데이터는 제거하였다. 그 결과, 실제 실험에서는 불완전 응답 데이터를 제외한 총 138명(A: 56명, B: 82명)의 데이터를 분석에 사용하였다.

실험은 A 중학교와 B 중학교를 구분하여 실시하였다. 그 이유는 두 학교에서 사용한 교과서가 상이하기 때문이다. 그리고, 패킷 요금 방식, 정액 요금 방식의 두 가지로 구분하여 다음과 같이 분석을 진행하였다.

- 패킷 요금 방식: 비용이 패킷 양에 의존적이므로 제 3.2절에서 설명한 바와 같이 선인출되는 데이터 양은 전송되는 패킷 크기 내에서 원하는 데이터 이외의 여분공간으로 한정된다. 따라서, 선인출할 수 있는 단어의 개수가 매우 제한적이 되므로, 실험에서는 선인출하는 단어 개수를 1, 2, 3의 적은 수로 한정하여 선인출 효과를 분석하였다.
- 정액 요금 방식: 비용이 패킷 양과 무관하므로 제3.2절에서 설명한 바와 같이 선인출되는 데이터 양은 모바일 메모리 양에 저장 가능한 최대량을 가져올 수 있다. 그러나, 일반적으로 교과서 지문의 새로 나온 단어의 수가 제한적이다(실제 각 단원의 새 단어 개수는 7~24개 정도이다). 따라서, 실험에서는 선인출하는 단어의 수를 패킷 요금 방식 보다 조금 더 많은 수인 5, 7, 9개로 변경하면서 분석하였다.

그리고, 선인출된 데이터는 모바일의 메모리에 캐싱되어 관리된다. 본 논문에서는 분석을 용이하게 하기 위하여, 바로 직전에 선인출된 단어들만이 캐싱된다고 가정하고 실험을 수행한다. 즉, 직전에 선인출된 단어에 의한 선인출 효과만을 분석하며, 과거에 선인출된 단어들

이 메모리 허용 범위에서 지속적으로 캐칭됨으로써 나타나는 성능 개선 효과는 분석에서 제외한다.

실험 방법으로는 선인출을 사용하는 경우와 그렇지 않은 경우(즉, 요구인출인 경우)를 비교하였다. 실험 결과에서는 선인출에 의한 인출 횟수를 *Prefetch*, 요구인출에 의한 인출 횟수를 *OnDemand*라 각각 표기한다. 실험 결과로는 사용자가 요청한 단어의 수에 대한 선인출 단어

개수의 비율인 선인출 비율($= \frac{OnDemand - Prefetch}{OnDemand} \times 100$)을 분석하고, 각 방법에 대해 모바일과 서버와의 라운드트립 횟수를 계산하였다.

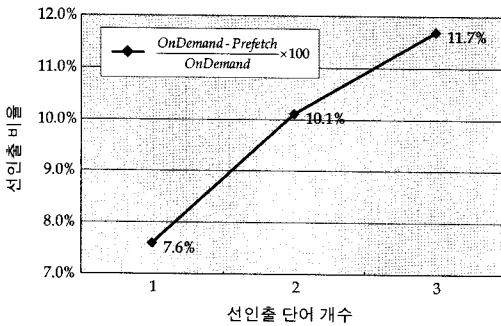
4.2 실험 결과

실험 1. 패킷 요금 방식의 실험 결과

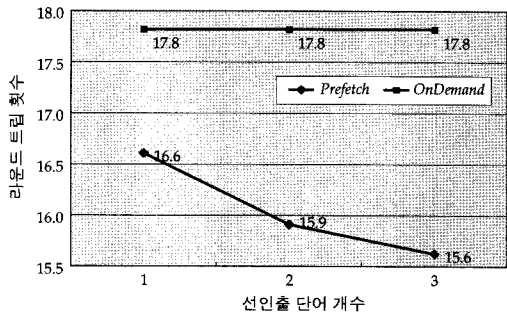
그림 6은 패킷 요금 방식의 경우 A 중학교에 대한 선인출 비율과 라운드트립 횟수를 나타내는 그래프이다. 그림 6(a)를 보면, 제한한 선인출 방법을 사용하면 대략 7.6%~11.7%까지 선인출 효과가 있음을 알 수 있다. 그리고, 선인출하는 단어의 개수가 증가함에 따라 선인출 비율이 증가함을 알 수 있다. 특히, 하나의 단어를 선인출하는 경우에도 평균 7.6%의 비교적 높은 성능 향상을 나타냈다. 이는 패킷 제한으로 인해 1~2개의 아주 적은

단어에 대한 선인출을 수행하더라도, 비교적 높은 선인출 효과를 볼 수 있음을 의미한다. 그림 6(b)를 보면 요구인출의 경우 선인출 단어 개수가 변화하여도 라운드트립 횟수가 일정한 값을 갖는다. 이는 모바일에서 단어를 검색할 때마다 서버에 단어 인출을 요청하기 때문이다. 반면에, 선인출의 경우 단어 개수가 증가함에 따라 라운드트립 횟수가 점차 감소함을 알 수 있다. 이는 선인출 방법의 경우 선인출 효과가 클수록 사용자가 요청한 단어의 횟수에 대한 서버에서 인출하는 단어의 횟수가 감소하기 때문이다. 이에 따라, 선인출의 라운드트립 횟수는 그림 6(a)의 선인출 비율 그래프와 반대로 감소하는 모양을 나타냄을 알 수 있다.

그림 7은 패킷 요금 방식의 경우 B 중학교에 대한 실험 결과를 나타낸다. 그림 7의 그래프 형태는 그림 6과 매우 유사함을 알 수 있다. 그러나, 그림 7(a)의 선인출 비율은 13.9%~25.2%로 그림 6(a)의 A 중학교의 실험 결과 치보다 높아진 것을 알 수 있다. 이러한 이유는 발췌한 지문에 새로 나온 단어의 수가 A 중학교에 비하여 B 중학교에 많을 뿐 아니라, 새로운 단어가 연속적으로 나타나는 비율이 B 중학교에서 더 높기 때문이다. 이러한 실험 결과를 볼 때, 새로 나온 단어의 수가 많고 연

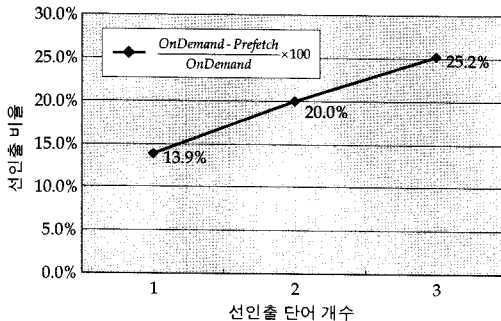


(a) 선인출 비율

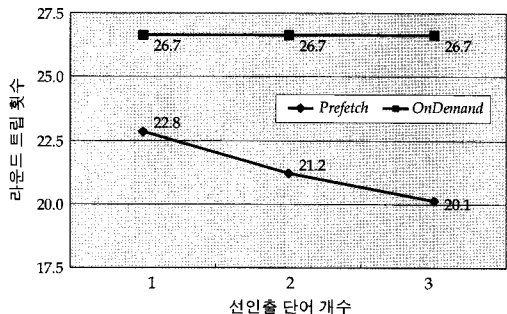


(b) 라운드 트립 횟수

그림 6 A 중학교의 패킷 요금 방식 실험 결과



(a) 선인출 비율



(b) 라운드 트립 횟수

그림 7 B 중학교의 패킷 요금 방식 실험 결과

속적으로 나타날수록 높은 선인출 효과를 기대할 수 있음을 알 수 있다.

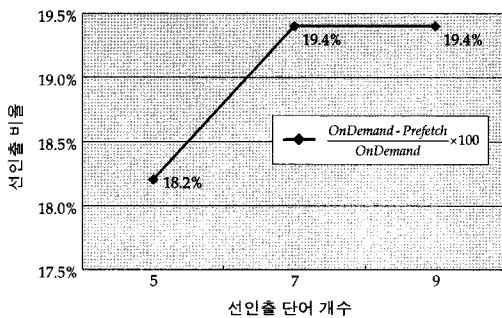
실험 2. 정액 요금 방식의 실험 결과

그림 8은 정액 요금 방식의 경우 A 중학교에 대한 선인출 비율과 라운드트립 횟수를 나타내는 그래프이다. 그림 8(a)를 보면, 선인출 단어의 개수가 5개에서 7개로 증가함에 따라 선인출 비율도 18.2%에서 19.4%로 증가함을 알 수 있다. 그런데 선인출 단어의 개수가 7개에서 9개로 증가했음에도 불구하고 선인출 비율은 19.4%로 동일한 값을 나타냈다. 이는 A 중학교 교과서의 새로 나온 단어의 개수가 7~9개로 매우 적어서 정액 요금의 경우에는 한 번의 선인출로 새로 나온 단어를 대부분이 모두 인출하기 때문이다. 그림 8(b)의 라운드트립 횟수를 보면, 패킷 요금 방식처럼 요구인출의 경우는 선인출 단어 개수와 상관없이 일정한 반면, 선인출의 경우 선인출 단어 개수가 증가함에 따라 횟수가 감소한 것을 알 수 있다. 또한, 선인출 단어 개수가 7개에서 9개로 증가하여도 라운드트립 횟수는 14.3%의 일정한 값을 유지했다. 이는 그림 8(a)에서와 같이 7개와 9개에서 선인출 비율이 변화하지 않기 때문이다.

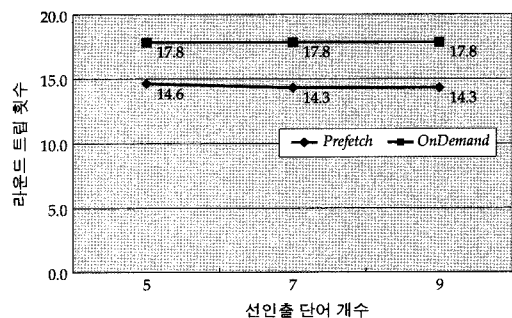
마지막으로, 그림 9는 정액 요금 방식의 경우 B 중학

교에 대한 실험 결과를 나타낸다. 그림 9(a)를 보면, 지금까지 살펴본 선인출 비율 그래프와 마찬가지로 단어의 개수가 증가함에 따라 선인출 비율이 증가함을 알 수 있다. 평균 선인출 비율은 31.0%~35.3%까지 높은 결과를 나타냈다. 그리고, 그림 9(b)의 라운드트립 횟수를 보면, 그림 8(b)와 마찬가지로 요구인출의 경우는 선인출 단어 개수와 상관없이 29.7%의 일정한 값을 나타낸 반면, 선인출의 경우는 선인출 단어 개수가 5에서 9개로 증가함에 따라 그 횟수가 18.8%에서 17.7%로 조금씩 감소한 것을 알 수 있다.

그림 8과 그림 9의 정액 요금 방식에 대한 실험에 있어서, 선인출하는 단어의 개수를 9개 이상으로 할 경우에는 보다 높은 선인출 효과를 얻을 수 있다. 즉, 새롭게 소개된 단어를 보다 많이 선인출할수록 보다 높은 선인출 효과를 나타내고, 이에 따라 성능을 향상시킬 수 있다. 그러나, 본 실험에서 사용한 교과서 지문의 경우 새로운 단어가 최대 9개였으며, 이에 따라서 본 논문의 실험에서는 최대 9개까지만 선인출하였다. 앞서 제3.2절에서 설명한 바와 같이 정액 요금 방식의 경우 선인출할 데이터의 양은 모바일 기기의 메모리 크기에 제한될 뿐이므로, 실제로는 9개 이상의 많은 단어를 선인출하는

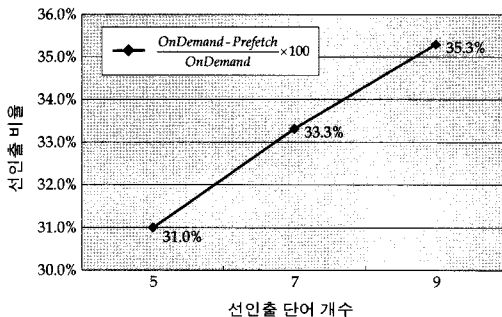


(a) 선인출 비율

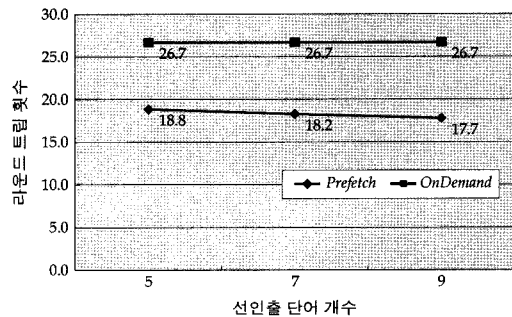


(b) 라운드 트립 횟수

그림 8 A 중학교의 정액 요금 방식 실험 결과



(a) 선인출 비율



(b) 라운드 트립 횟수

그림 9 B 중학교의 정액 요금 방식 실험 결과

것이 가능하다. 따라서, 새로운 단어가 많이 나오는 교과서의 다른 지문이나 다른 교과서를 사용한다면 보다 많은 수의 단어를 한꺼번에 선인출하는 방법을 사용할 수 있으며, 이 경우 보다 높은 성능 개선을 기대할 수 있다.

지금까지의 두 중학교에 대한 두 가지 요금 방식에 대한 실험 결과를 요약하면 다음과 같다.

- 패킷 요금 방식의 경우, 제안한 선인출 방법은 기존 요구인출에 비해 평균 9.8%~19.7%의 선인출 비율을 나타냈다. 특히, 단지 하나의 새로운 단어에 대한 선인출을 수행하더라도 평균 7.6%~13.9%의 비교적 높은 선인출 비율을 나타냈다.
- 정액 요금 방식의 경우, 제안한 선인출 방법은 기존 요구인출에 비해 평균 19.0%~33.2%의 높은 선인출 비율을 나타냈다.

결국, 제안한 방법은 한 개에서 수 개의 단어에 대한 선인출을 통해 10% 이상의 높은 선인출 비율을 나타내고, 이를 통해 라운드트립 횟수를 크게(10% 이상) 줄일 수 있어, 모바일 사전에 적합한 우수한 선인출 기법이라 할 수 있다.

5. 결론

본 논문에서는 모바일 사전의 성능을 향상시키기 위한 선인출 방법을 제안하였다. 현재까지의 모바일 관련 연구는 콘텐츠 및 서비스를 유선 환경에서 모바일 환경으로 옮기는데 연구의 초점이 맞추어져 있었다. 반면에, 본 연구에서는 선인출 기법을 모바일 사전에 적용하여, 네트워크 비용을 절감하고 통신 효율을 높이는 방법을 제안하였다.

- 선인출 기반 모바일 사전의 실현을 위해 본 논문에서 제안한 내용은 다음과 같이 요약할 수 있다. 첫째, 모바일 사전에 선인출 기법을 적용한 동작 프레임워크를 제안하였다. 둘째, 모바일 사전에 선인출 기법을 적용하기 위하여 선인출할 데이터 양을 패킷 요금 방식과 정액 요금 방식으로 구분하여 제안하였다. 셋째, 선인출 데이터를 결정하는 것은 응용의 종류와 사용자의 프로파일에 따라 달라질 수 있으므로, 본 논문에서는 중·고등학생을 대상으로 한 모바일 영한사전에 초점을 맞추어 선인출 데이터를 결정하였다. 즉, 선인출할 데이터는 연속적으로 액세스될 가능성이 높은 중·고등학교의 새로 나온 단어를 대상으로 하여 선인출 비율의 효과를 높였다. 넷째, 제안한 데이터 양 및 데이터 종류 결정 방법을 바탕으로 모바일 사전을 위한 선인출 알고리즘을 제안하였다. 다섯째, 실험을 통하여 기존 요구인출에 비해 패킷 요금 방식과 정액 요금 방식의 경우 각각 평균 9.8%~19.7%, 평균 19.0%~33.2%의 높은 선인출 비율을 나타냈다. 특히,

한두 개의 단어 만을 선인출하더라도 10% 내외의 비교적 높은 선인출 비율을 보임으로서 이동통신 기반의 패킷 요금 방식에 사용될 수 있는 실용성을 입증하였다. 또한, 더욱 많은 단어를 선인출할수록 더욱 높은 선인출 비율을 보임을 확인하여, 무선 인터넷과 같은 정액 요금 방식에서 더더욱 효과적으로 사용될 수 있음을 확인하였다. 이 같은 결과를 볼 때, 본 논문의 결과는 모바일 사전뿐 아니라 선인출 기법을 필요로 하는 무선 인터넷 기반의 여러 응용에 적용될 수 있는 우수한 연구 결과라 사료된다.

참고 문헌

- [1] jDictionary Mobile, <http://jdictionary-mobile.com/index.html>
- [2] SABC Mobile, <http://www.sabcmobile.co.za/mobidic.htm>
- [3] Windows Mobile, <http://windows-mobile.qarchive.org/index.html>
- [4] Han, W.-S., Whang, K.-Y., and Moon, Y.-S., "A Formal Framework for Prefetching Based on the Type-Level Access Pattern in Object-Relational DBMSs," *IEEE Trans. on Knowledge and Data Engineering*, Vol.17, No.10, pp. 1436-1448, Oct. 2005.
- [5] Bernstein, P. A., Pal, S., and Shutt, D., "Context-Based Prefetch for Implementing Objects on Relations," In *Proc. of the 21st Int'l Conf. on Very Large Data Bases*, Edinburgh, Scotland, pp. 327-338, Sept. 1999.
- [6] Nanopoulos, A., Katsaros, D., and Manolopoulos, Y., "A Data Mining Algorithm for Generalized Web Prefetching," *IEEE Trans. on Knowledge and Data Engineering*, Vol.15, No.5, pp. 1155-1169, Sept./Oct. 2003.
- [7] Xu, C.-Z., and Ibrahim, T. I., "A Keyword-Based Semantic Prefetching Approach in Internet News Services," *IEEE Trans. on Knowledge and Data Engineering*, Vol.16, No.5, pp. 601-611, May 2004.
- [8] Hua, Z. et al., "Design and Performance Studies of an Adaptive Scheme for Serving Dynamic Web Content in a Mobile Computing Environment," *IEEE Trans. on Mobile Computing*, Vol.5, No.12, pp. 1650-1662, Dec. 2006.
- [9] Laakko, T. and Hiltunen, T., "Adapting Web Content to Mobile User Agents," *IEEE Internet Computing*, Vol.9, No.2, pp. 46-53, Mar./Apr. 2005.
- [10] Lemlouma, T. and Layaïda, N., "Content Interaction and Formatting for Mobile Devices," In *Proc. of ACM Symp. on Document Engineering*, pp. 98-100, Bristol, United Kingdom, Nov. 2005.
- [11] Padmanabhan, V. N. and Mogul, J. C., "Using Predictive Prefetching to Improve World Wide Web Latency," *ACM SIGCOMM Computer Com-*

munication Review, Vol.26, Issue 3, pp. 22-36, July 1996.

- [12] Jiang, Z. and Kleinrock, L., "An Adaptive Network Prefetch Scheme," *IEEE Journal on Selected Areas in Communications*, Vol.16, No.3, pp. 358-368, Apr. 1998.
- [13] Yan, C., Shen, J., and Peng, Q., "Parallel Web Prefetching on Cluster Server," In *Proc. of IEEE Canadian Conf. on Electrical and Computer Engineering*, pp. 2284-2287, May 2005.
- [14] Park, S.-Y., Park, D., Lee, J., and Cho, J.-W., "Efficient Inter-Backend Prefetch Algorithms in Cluster-based Web Servers," *HPC Asia 2001*, pp. 24-28, Queensland, Australia, Sept. 2001.
- [15] Kim, W. et al., "Architecture of the ORION Next-Generation Database System," *IEEE Trans. on Knowledge and Database Engineering*, Vol.2, No.1, pp. 109-124, Mar. 1990.
- [16] Palmer, Z. and Zdonik, S. B., "Fido: A Cache That Learns to Fetch," In *Proc. of the 17th Int'l Conf. on Very Large Data Bases*, Barcelona, Spain, pp. 255-264, Sept. 1991.
- [17] Chang, E. E. and Katz, R. H., "Exploiting Inheritance and Structure Semantics for Effective Clustering and Buffering in an Object-Oriented DBMS," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, Portland, Oregon, pp. 348-357, May 1989.
- [18] Bernstein, P. A., Pal, S., and Shutt, D., "Context-Based Prefetch for Implementing Objects on Relations," In *Proc. of the 21st Int'l Conf. on Very Large Data Bases*, Edinburgh, Scotland, pp. 327-338, Sept. 1999.



홍순정

2004년 2월 강원대학교 컴퓨터과학과 학사. 2007년 2월 강원대학교 교육대학원 컴퓨터교육전공 석사. 2006년 3월~2007년 2월 한국과학기술원 첨단정보기술연구원 연구보조원. 관심분야는 Computer Education, Data Mining, Knowledge

Discovery, Data Warehousing



문양세

1991년 2월 한국과학기술원 과학기술대학 전산학과 학사. 1993년 2월 한국과학기술원 전산학과 석사. 2001년 8월 한국과학기술원 전자전산학과 전산학전공 박사. 1993년 2월~1997년 2월 현대전자산업(주) 통신사업본부 주임연구원. 2001년 9월~2002년 2월 (주)현대시스콤 호처리개발실 선임연구원. 2002년 2월~2005년 2월 (주)인프라밸리 기술연구소 기술위원(이사). 2005년 3월~현재 한국과학기술원 첨단정보기

술연구센터 연구원. 2005년 3월~현재 강원대학교 컴퓨터과학과 조교수. 관심분야는 Data Mining & Knowledge Discovery, Stream Data, Database Security & Privacy, Storage System, Database Applications, Mobile/ Wireless Communication Services & Systems



김혜수

2003년 2월 강원대학교 컴퓨터정보통신공학부 학사. 2006년 2월 강원대학교 교육대학원 컴퓨터교육전공 석사. 2006년 3월~현재 강원대학교 컴퓨터과학전공 박사과정. 2005년 3월~현재 한국과학기술원 첨단정보기술연구원 연구원. 관심분야는 Data Mining & Knowledge Discovery, Database Security & Privacy, Computer Education



김진호

1982년 2월 경북대학교 전자공학과 학사. 1985년 2월 한국과학기술원 전산학과 석사. 1990년 2월 한국과학기술원 전산학과 박사. 1995년 8월~1996년 7월 미국 미시간 대학교 객원 교수. 2003년 2월~2004년 2월 미국 Drexel University 객원 교수. 1999년 3월~현재 한국과학기술원 첨단정보기술연구원 연구원. 1990년 8월~현재 강원대학교 컴퓨터과학과 교수. 2006년 9월~현재 강원대학교 중앙교육연구전산원 원장. 관심분야는 Data warehouse, OLAP, Data Mining, Real-time/Embedded Database, Main-memory database, Data Modeling, Web Database Technology



정영준

1974년 2월 서울대학교 전기공학과 학사. 1984년 5월 미국 university of Kansas 컴퓨터공학 석사. 1988년 10월 미국 university of Kansas 컴퓨터공학 박사. 1974년 6월~1981년 1월 국방과학연구소 연구원. 1982년 3월~1988년 5월 미국 Kansas Geological Survey Computer Center 연구원. 1988년 5월~2001년 8월 미국 RANPAC R&D Center 책임연구원. 2001년 3월~2004년 9월 강원대학교 중앙교육연구전산원 원장. 2004년 8월~2005년 8월 미국 California State University at Pomona 객원교수. 1991년 8월~현재 강원대학교 컴퓨터과학과 교수. 관심분야는 Wireless & Mobile Communications Systems, Wireless Sensor Networks, Internet Applications & Services, Network Security, Network Management