주제기반 모바일 웹 콘텐츠 적응화
(Topic-Specific Mobile Web Contents Adaptation)

이 은 실 † 강 진 범 †† 최 중 민 †††
(Eunshil Lee) (Jinbeom Kang) (Joongmin Choi)

요 약 모바일 콘텐츠 적응화는 데스크탑 PC 용으로 제작되고 표현된 웹 콘텐츠를 크기와 정보량이 제한된 사용자의 무선 모바일 디바이스 환경에 맞게 변환하여 표현해주는 적응화 기술을 말한다. 기존의 웹 콘텐츠 적응화 방법은 대부분 장치 의존적인 접근 방법을 취했다. 또한 소형 장치에 맞게 콘텐츠를 변환하는 작업이 대부분 수동으로 이루어졌고 콘텐츠와 연관된 문맥 정보가 제공되지 않았다. 이 외에도 사용자의 선호도를 반영하지 못하여 모든 사용자에게 동일한 정보를 제공하였다. 이와 같이 기존의 모바일 콘텐츠 적응화 방법은 범용성, 확장성, 사용자 적응성에 문제가 있었고, 그 결과 사용자는 방대한 양의 콘텐츠 중에서 자신이 원하는 정보를 선택하는데 어려움을 겪을 수밖에 없었다.

이러한 문제점을 해결하기 위해 본 논문에서는 모바일 디바이스에 대한 새로운 웹 콘텐츠 적응화 기법을 제시한다. 제안하는 기법의 특징은 모바일 디바이스 적응화와 사용자 적응화를 동시에 적용하는 자동화된 콘텐츠 적응화를 시도하였다는 것이다. 이를 위해 웹 콘텐츠 적응화 과정을 블록 필터링, 블록 제목 추출, 블록 콘텐츠 요약, 학습을 통한 개인화 등의 4 단계로 구성하였다. 이러한 과정을 통해 웹페이지를 블록 단위로 나눠서 불필요한 블록을 제거하고 사용자가 필요로 하는 콘텐츠 블록만을 선별하여 모바일 디바이스에 나타내며, 학습을 통해 사용자가 관심을 가지는 정보를 정보목록의 상위에 놓음으로써 사용자가 선호정보를 편리하게 사용할 수 있도록 하였다. 온라인 뉴스사이트를 서점을 대상으로 한 일련의 실험을 통해 제안하는 모바일 웹 콘텐츠 적응화의 성능을 평가하였으며 디바이스 적응화와 사용자 적응화 모두 만족한 결과를 얻을 수 있었다.

키워드 : 모바일 콘텐츠 적응화, 웹 콘텐트 적응화, 웹 콘텐츠 마이닝

Abstract Mobile content adaptation is a technology of effectively representing the contents originally built for the desktop PC on wireless mobile devices. Previous approaches for Web content adaptation are mostly device-dependent. Also, the content transformation to suit to a smaller device is done manually. Furthermore, the same contents are provided to different users regardless of their individual preferences. As a result, the user has difficulty in selecting relevant information from a heavy volume of contents since the context information related to the content is not provided.

To resolve these problems, this paper proposes an enhanced method of Web content adaptation for mobile devices. In our system, the process of Web content adaptation consists of 4 stages including block filtering, block title extraction, block content summarization, and personalization through learning. Learning is initiated when the user selects the full content menu from the content summary page. As a result of learning, personalization is realized by showing the information for the relevant block at the top of the content list. A series of experiments are performed to evaluate the content adaptation for a number of Web sites including online newspapers. The results of evaluation are satisfactory, both in block filtering accuracy and in user satisfaction by personalization.

Key words : mobile contents adaptation, Web contents adaptation, Web contents mining

· 이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(과제번호 KRF-2005-041-D00692)
† 정 회 원 : LG전자 MC단말연구소 연구원
lovelus@gmail.com
†† 학생회원 : 한양대학교 컴퓨터공학과
jbkang@cse.hanyang.ac.kr
††† 종신회원 : 한양대학교 컴퓨터공학과 교수
jmchoi@cse.hanyang.ac.kr
논문접수 : 2006년 8월 7일
심사완료 : 2007년 4월 9일

1. 서 론

오늘날 인터넷과 더불어 휴대전화나 PDA와 같은 다양한 무선 모바일 장치들이 널리 보급되고 있다. 이러한 모바일 디바이스 사용자들은 데스트탑 PC에서 제공받았던 풍부하고 다양한 콘텐츠와 서비스를 모바일 장치에서도 받아볼 수 있기를 원한다. 하지만 모바일 디바이스마다 서로 다른 기능과 사용자 선호도, 네트워크 대여폭 때문에 서비스는 제한될 수밖에 없다. 더욱이 모바일 디바이스는 화면 크기가 제한되어 있고 또한 기기의 경량화를 위해 다양한 콘텐츠를 보여주기 위한 모듈과 같은 부수적인 기능을 제거하기 때문에 일반 PC에서 폭넓게 사용하고 있는 웹 정보를 모바일 사용자가 불편없이 열람하기 위해서는 콘텐츠 적응화(contents adaptation)라는 과정이 필요하다.

웹에서 제공되는 콘텐츠는 대부분 모바일 장치에서 바로 볼 수 없다. 이것은 웹 콘텐츠가 모바일 디바이스에 표현되기에는 방대한 내용을 담고 있거나, 표현할 수 없는 형식으로 제작되었거나, 또는 음성이나 플래시와 같은 모바일 장치에서 지원하지 않는 멀티미디어 정보를 담고 있는 경우 등이다. 모바일 콘텐츠 적응화는 이러한 다양한 형태의 콘텐츠를 모바일 디바이스에 맞게 효과적으로 보여주는 방법이다. 좀 더 구체적으로 표현하면 PC용으로 제작되고 표현된 웹 콘텐츠를 크기와 정보량이 제한된 사용자의 무선 모바일 디바이스 환경에 맞게 변환하여 표현해주는 적응화 기술을 말한다.

기존의 웹 콘텐츠 적응화 방법은 대부분 장치 의존적인 접근 방법을 취했다. 또한 소형 장치에 맞게 콘텐츠를 변환하는 작업이 대부분 수동으로 이루어졌고 콘텐츠와 연관된 문맥 정보가 제공되지 않았다. 이 외에도 사용자의 선호도를 반영하지 못하여 모든 사용자에게 동일한 정보를 제공하였다. 이와 같이 기존의 모바일 콘텐츠 적응화 방법은 확장성과 사용자 적응성에 문제가 있었고, 그 결과 사용자는 방대한 양의 콘텐츠 중에서 자신이 원하는 정보를 선택하는데 어려움을 겪을 수밖에 없었다.

이러한 문제점을 해결하기 위해 본 논문에서는 모바일 디바이스에 대한 새로운 웹 콘텐츠 적응화 기법을 제시한다. 제안하는 기법의 특징은 모바일 디바이스 적응화와 사용자 적응화를 동시에 적용하는 자동화된 콘텐츠 적응화를 시도하였다는 것이다. 이를 위해 웹 콘텐츠 적응화 과정을 블록 필터링, 블록 제목 추출, 블록 콘텐츠 요약, 학습을 통한 개인화 등의 4 단계로 구성하였다. 모바일 디바이스 적응화는 처음 3단계인 블록 필터링, 블록 제목 추출, 블록 콘텐츠 요약의 과정을 거쳐 이루어지고, 사용자 적응화는 마지막 단계인 학습을 통한 개인화 과정을 통해 이루어진다. 이러한 모든 적응화 과정은 특정 디바이스의 하드웨어 정보와는 무관하게 진행되므로 범용성과 확장성을 보장한다. 또한 개인화 단계를 통해 사용자 선호를 정보를 목록의 상위에 위치시킴으로써 사용자 적응성을 지원하고 사용자의 신속한 정보접근을 가능하게 한다.

본 논문에서 제안하는 디바이스 적응화의 핵심 기법은 시각적 웹페이지 세그멘테이션 기법을 웹 콘텐츠 변환에 적용하는 것이다. 즉, 웹페이지를 사용자가 시각적으로 구분하는 것과 유사한 블록으로 나누고, 각 블록의 속성을 파악하여 필요한 블록과 불필요한 블록을 인식한다. 실제 내용을 나타내는 블록의 주제를 추출하여 모바일 장치에 제공하는 효율적인 콘텐츠 적응화 기법을 제시한다. 또한, 사용자 적응화의 핵심은 개인화된 콘텐츠를 제공하기 위해 적응화 과정에서 학습을 기반으로 사용자가 선호하는 정보만을 빠르게 제공할 수 있는 기법을 제시한 것이다. 사용자가 선택한 콘텐츠의 단어들을 기반으로 사용자의 선호주제를 인식하는 간단한 학습 기법을 이용하여 사용자가 관심을 가지는 정보를 정보목록의 상위에 놓음으로써 사용자가 선호정보를 편리하게 사용할 수 있도록 하였다.

온라인 뉴스사이트를 서점을 대상으로 한 일련의 실험을 통해 제안하는 모바일 웹 콘텐츠 적응화의 성능을 평가하였으며 디바이스 적응화와 사용자 적응화 모두 대체로 만족한 결과를 얻을 수 있었다.

본 논문은 다음과 같이 구성된다. 1장에서는 모바일 콘텐츠 적응화의 개념과 기존 방법의 문제점, 본 시스템의 개선 사항 등을 설명하였다. 2장에서는 기존의 관련 연구를 요약하고 문제점을 구체적으로 분석한 다음 본 논문의 접근방향을 제시한다. 3장은 본 논문에서 제안하는 모바일 웹 콘텐츠 적응화 기법을 단계별로 기술하고 예제를 통해 설명한다. 4장은 제안하는 기법의 성능을 평가하기 위한 실험 환경과 평가 결과를 기술한다. 5장에서는 결론과 향후 연구방향을 제시한다.

2. 관련 연구

모바일 콘텐츠 적응화란 PC기반에서 만들어진 콘텐츠를 무선 디바이스에 효과적으로 표현하기 위한 기술이다. 모바일 디바이스 사용자들은 PC에 제공되는 정보와 같이 풍부하고 다양한 콘텐츠를 요구하고 있지만 이질적인 환경인 클라이언트 디바이스는 각각 다른 기능, 사용자 선호도, 네트워크 대역폭 등을 가지고 있고 그에 따른 제한된 서비스만을 제공 받고 있다.

URICA(Usage-awaRe Interactive Content Adaptation) [1]는 Mohomed가 제안한 시스템으로, 사용자의 특성에 맞게 콘텐츠가 제공될 수 있도록 사용자의 문맥 정보(device type, screen size, network bandwidth, user

location)를 기반으로 콘텐츠를 어떻게 적용할지 학습한다. 사용자들의 센서를 통해 문맥 정보를 수집하고 유사한 정보를 가진 다른 사용자들로부터 어떻게 콘텐츠를 제공할 것인지 결정한다. 이러한 작업을 위해 유사한 문맥 정보를 가진 사용자들끼리 그룹핑을 하고 사용자가 선호하는 형태의 정보가 제공되도록 지속적으로 학습하는 FCS(Feedback-driven Context Selection) 모듈을 기반으로 적응화가 진행된다.

하지만 콘텐츠 적응화를 위해 문맥을 학습하고 그룹핑을 통해 사용자가 선호하는 콘텐츠 형태를 예측하지만 개개인 성향 및 관심분야를 고려해 콘텐츠 정보를 적응화가 이루어지지 않고 있다. 단순히 장치 종류(device type)와 같은 문맥 정보를 기반으로 어떻게 콘텐츠 정보를 표현할지에 중점을 두고 웹 문서 상의 방대한 정보를 filtering 없이 제공한다.

Blekas[2]는 콘텐츠 적응화의 효율을 높이기 위해 RSS feed를 이용하였다. 일반 웹 문서를 사용자에게 RSS feed의 <title>, <link>, <description>을 이용해 요약된 정보로 제공한다. 하지만, RSS feed가 포함되어 있지 않는 웹 문서의 경우 먼저 DOM 트리를 생성하고 <script>, <style>, <iframe>, <object>와 같은 불필요한 노드를 제거한다. 그런 다음 트리를 문서로 변환을 하고 크기가 10k를 넘는지 확인을 한다. 만약 10k를 넘는 문서가 형성되었다면, 내부 트리(sub-tree)로 분리하여 하나의 웹 문서를 여러 문서로 구성한다.

이와 같은 작업은 콘텐츠의 연관성 없이 문서를 나누어 제공함으로써 사용자에게 일관성 없는 정보를 제공할 수 있다. 더불어 RSS feed가 없는 웹 문서의 경우 사용자에게 단순한 형태의 필터링 및 마이닝 없이 정보를 제공한다. 그래서 사용자에게 불필요하고 광고와 같은 자료까지 무분별하게 사용자에게 제공될 수 있다.

Lam[3]이 제안한 Summary Thumbnails은 일반 웹 문서를 PDA와 같은 모바일 장치(mobile device)에 정제 없이 보여주면 사용자가 읽을 수 없는 형태로 표현된다는 것을 보였다. 그래서 웹 문서를 요약한 결과를 보여주고 사용자가 화면상에서 보고자하는 부분을 선택하면 선택된 부분이 확대되어 보여주는 형태를 취하고 있다. 하지만 요약된 정보가 명확한 문장의 형태를 갖추고 있지 않아 왜곡된 정보를 사용자에게 제공할 수 있다. 더불어 다양한 정보를 담고 있는 웹 문서의 레이아웃을 그대로 보여주는 것은 사용자가 한 눈에 문서의 내용을 파악하는데 무리가 있다. 그리고 사용자 선호 정보를 고려하지 않아 원하는 정보를 찾아 열람하는 것은 다소 사용자에게 불편함을 안겨줄 수 있다.

위의 세가지 관련연구를 통하여 기존의 모바일 콘텐츠 적응화 방법의 문제점은 대략 3가지 정도로 요약할 수 있다. 첫째, 대부분의 적응화 방법이 모바일 장치의 하드웨어 사양에 의존적이다[4]. 따라서 콘텐츠 적응화 시스템이 장치의 프로파일 정보를 알아야 하고 따라서 모든 모바일 디바이스에 대한 사양 정보를 가지고 있어야 한다. 결과적으로 범용적인 적응화 시스템이 구성되기 어렵고 확장성도 떨어진다. 둘째, 소형 모바일 장치에 맞게 콘텐츠를 변환하는 작업이 대부분 수동으로 이루어진다. 일부 웹사이트나 웹포털(네이버, 다음, 구글 등)에서 PDA에서 접근했을때 보여주기 위한 별도의 URL를 유지하여, 소형기기에 디스플레이 되도록 수동으로 미리 변형된 콘텐츠를 저장하는 것이 예가 될 수 있다. 따라서 이렇게 별도의 사이트를 가지고 있지 않은 일반 웹사이트의 콘텐츠는 모바일 디바이스에서 사용하기가 거의 불가능하다. 또한 일부 시스템은 웹페이지의 링크 정보만을 디바이스에 보여주는 형태를 취함으로써 콘텐츠와 연관된 문맥 정보가 제공되지 않고 그 결과 사용자는 방대한 양의 콘텐츠 중에서 자신이 원하는 정보를 선택하는데 어려움을 겪을 수밖에 없다[5,6]. 셋째, 웹사이트에 있는 콘텐츠를 있는 그대로 정렬하여 모든 사용자에게 동일한 형태의 콘텐츠를 제공한다. 따라서 만일 사용자가 원하는 콘텐츠가 리스트의 가장 아래에 위치하면 스크롤을 이용하여 페이지를 아래나 다른 페이지로 이동해야 한다. 즉, 각 사용자의 개인선호 정보를 이용하지 못하기 때문에 특정 사용자가 원하는 정보를 목록의 상위에 보여주지 못하고, 결국 사용자가 유용한 정보를 신속하게 접근하는 것을 지원하지 못한다.

위의 문제점을 해결하기 위해서 본 논문에서는 다음과 같은 개선된 방법을 제안한다. 첫째, 디바이스의 프로파일 정보에 국한되지 않고 모든 디바이스에 만족하는 적응화된 정보를 제공한다. 둘째, 웹페이지를 블록으로 나누어서 블록내의 정보를 표현하여 디바이스에 보여준다[7,8]. 기존에는 링크 주변에 무슨 내용이 존재하는지 알 수 없었고 사용자의 선택에 도움을 줄 수 있는 것은 링크되어있는 제목이나 내용 뿐이였다[9,10]. 그러나 블록을 기반으로 블록 내에서 제목추출과 내용을 요약하여 링크 주변에 내용을 알 수 있으므로 사용자가 원하는 정보를 선택적으로 취할 수 있다. 셋째, 사용자가 한번 본 내용을 학습하여 사용자가 관심을 가지고 있는 콘텐츠들을 리스트의 맨 상위로 이동시킨다. 결과적으로 특정 사용자가 선호하는 정보를 빠르게 얻을 수 있도록 편의를 제공한다.

3. 학습을 통한 모바일 콘텐츠 적응화

3.1 적응화 과정

모바일 콘텐츠 적응화 과정을 구현한 시스템 구조는 그림 1과 같이 모바일 디바이스(Client), 적응화 정을 수

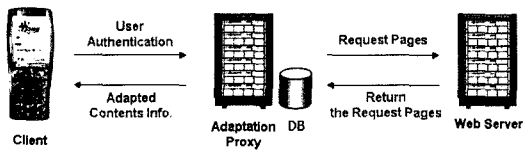

그림 1 적응화를 위한 시스템 구조

행하기 위한 프록시서버(Adaptation Proxy), 웹서버(Web Server)로 나누어진다.

사용자는 디바이스에 맞는 적응화된 콘텐츠를 취하기 위해 디바이스에서 본인임을 인증하기위한 정보를 프록시서버에게 보낸다. 프록시서버는 전송받은 정보와 데이타베이스에 저장된 사용자 인증 정보를 비교하여 서버의 사용 권한을 확인한다. 인증과정을 거치면 사용자는 본인이 원하는 웹페이지 주소를 입력하여 서버에게 전송한다. 프록시서버는 디바이스를 통해 전송받은 웹페이지 주소가 현재 캐시에 저장되어 있는 페이지인지를 확인한다. 캐시에 저장된 웹페이지라면, 이미 적응화된 웹페이지의 정보를 디바이스에 전송한다. 그렇지 않다면, 프록시서버는 웹서버에 웹페이지에 대한 정보를 요청한다. 프록시서버는 해당 웹서버를 통해 받은 웹페이지를 디바이스에 적절한 형태로 적응화 과정을 수행한 다음 사용자의 디바이스에 전송한다.

프록시 서버에서 수행하는 적응화 과정은 그림 2와 같이 4단계로 나누어진다. 우선 기존에 제안된 시각적 기반의 페이지 세그멘테이션 기법인 VIPS를 사용하여 웹페이지를 블록으로 나누었다고 가정한다.

첫 번째 단계에서는 블록 필터링을 통해서 네이게이션 바, 네비게이션 리스트, 콘텐츠와 같이 필요한 블록은 남겨두고 광고, 카피라이트 등 필요 없는 블록은 제거한다.

두 번째와 세 번째 단계는 제목추출과 내용요약이다. 콘텐츠 블록에서 제목을 추출하고 추출된 제목을 기반으로 콘텐츠의 내용을 요약한다. 위와 같은 과정을 통해 모바일 디바이스에는 네비게이션 바, 네비게이션 리스트, 콘텐츠 블록의 정보가 웹페이지에 나타난 순서대로 정렬된다. (이 3가지 블록 카테고리에 대해서는 다음 절에서 상세히 설명한다.) 콘텐츠 블록의 경우, 추출된 제목에 링크가 존재하고 링크를 선택하면 요약된 정보가 보여진다.

네 번째 단계는 학습단계이다. 사용자가 선택한 블록에서 빈도수가 높은 단어들을 추출하고 추출된 단어들의 가중치를 조절한다. 이를 통해 향후 사용자가 관심을 가지는 내용의 블록이 상위에 나타나 블록의 우선순위가 변경된다.

이 4단계 각각에 대한 상세한 설명을 다음에서 하고자 한다.

3.2 블록 필터링

블록 필터링 단계는 VIPS 알고리즘을 통해 나누어진 블록들을 휴리스틱한 규칙을 이용하여 콘텐츠(Contents) 블록, 네비게이션 바(Navigation Bar) 블록, 네비게이션 리스트(Navigation List) 블록의 세 가지 카테고리로 블록을 나누고 불필요한 블록은 제거하는 과정이다. 콘텐츠 블록은 내용을 포함하는 블록이고, 네비게이션 바 블록은 메뉴를 포함하고 있는 블록이며, 네비게이션 리스트 블록은 내용을 포함할 가능성이 있는 링크들을 가지고 있는 블록이다. 그림 3의 왼쪽은 예제 웹 페이지에서 블록들이 3가지 카테고리로 분류된 결과를 보여준다.

그림 4는 블록의 카테고리 분류 규칙을 보여준다. 먼저, 카테고리를 분류하기 전에 VIPS에 의해 나눠진 블록의 내용이 전체 페이지의 50% 이상을 차지하는 경우에만 내용을 포함할 가능성이 있는 블록으로 간주하여 블록 카테고리 분류가 시작된다. 여기에 해당되는 블록은 그림 4의 규칙에 따라 콘텐츠 블록, 네비게이션 바 블록, 네비게이션 리스트 블록으로 분류된다. 예를 들어, 네비게이션 바 블록으로 분류되기 위해서는 블록의 링

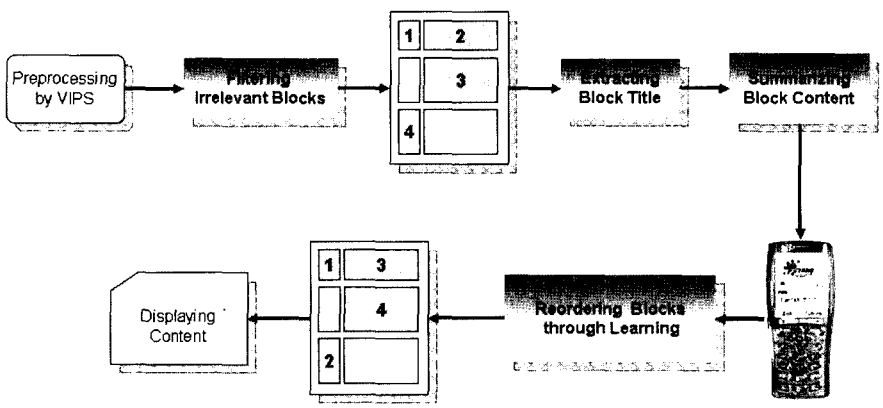

그림 2 적응화 과정

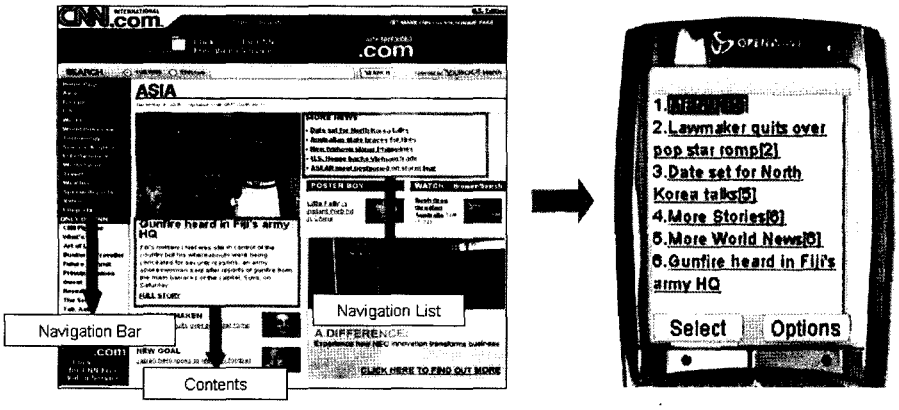

그림 3 웹페이지를 카테고리 분류한 결과

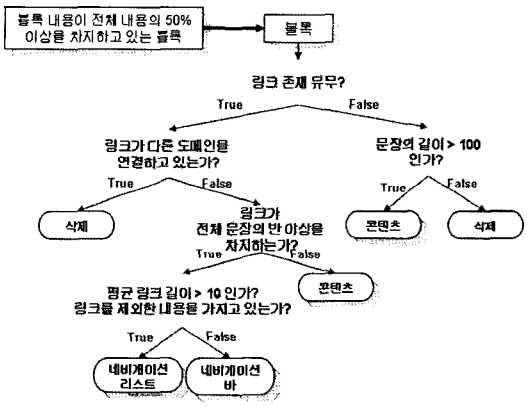

그림 4 블록 카테고리 분류 규칙

크가 존재하고('링크 존재 유무?'가 True), 링크가 다른 사이트로 연결되지 않고('링크가 다른 도메인을 연결하고 있는가?'가 False), 링크가 전체 문장의 반 이상을 차지하고, 링크를 제외한 내용을 가지고 있지 않을 경우이다. 네비게이션 리스트 블록이나 콘텐츠 블록도 같은 방법으로 인식할 수 있다. 만일 '링크 존재 유무'가 True이고 '링크가 다른 도메인을 연결하고 있는가?'가 True이면 광고일 가능성이 높으므로 해당 블록은 제거된다.

앞에서 본 그림 3의 오른쪽 그림은 블록 필터링 단계를 수행한 후 예제 웹 페이지 내용이 모바일 디바이스에 표시된 결과이다. 웹페이지에서 붉은색 테두리로 표시된 3개의 블록 카테고리들이 모바일 디바이스에서 메뉴(파란색으로 표시), 콘텐츠(어두운 붉은 색으로 표시), 네비게이션 리스트(제목과 괄호안의 리스트의 개수로 표시)로 정확하게 분류되었음을 확인할 수 있다.

3.3 제목 추출

제목추출(Title Extraction)은 주어진 문서에서 내용을 대표할 수 있는 단어의 집합이나 구(Phrase)를 인식하고 추출하는 것이다. 제목이나 대표단어 추출에 대한 기존 연구를 요약하면 다음과 같다. WebLearn[8]은 구글(google)을 통해서 관심있는 정보에 대한 질의를 입력하면 검색된 웹문서에서 불필요한 문서는 제거하고 관련된 문서에서 질의에 대한 하위 개념이 될 수 있는 단어나 구를 추출한다. WebLearn에서는 관련문서에서 <h1> <h2> <h3> <h4> <b> <big> <i> <u> <dt> 와 같은 태그정보가 있을 경우 문서의 중심이 되는 단어나 구로 판단한다. [9]의 연구에서는 HTML 문서에서 나타나는 제목의 특징(features)을 정의하고 확률적인 모델링을 이용해서 수치화한다. 그 후 수치화한 특징들을 입력 신호로 하여 학습을 통해 가중치를 설정하고, 이렇게 학습으로 설정된 가중치를 바탕으로 제목을 추출한다. 이 연구에서 가정한 HTML 문서에서 나타나는 제목의 특징으로는 문서에서 나타날 수 있는 제목의 수, 위치, 제목의 형태, 제목의 내용정보 등이 있다. [12]의 연구에서는 주어진 문장에서 제목을 생성하는 방법이 제안되었다. 이 방법은 stop words, cue words, cliche words를 이용하여 주요 단어 및 문장을 선별하고 이들의 특성을 고려하여 주제를 잘 나타내는 절을 선택한다.

본 논문에서는 블록 필터링을 통해 분류된 블록 중 콘텐츠 블록에서 제목을 추출한다. 본 논문에서 주로 모바일 디바이스에 표현하고자 하는 내용은 뉴스 사이트의 정보이다. 뉴스 정보를 나타내는 웹페이지는 태그들로 구성되어 있으며 따라서 WebLearn에서 사용된 것과 유사하게 HTML의 태그 정보를 이용하여 뉴스 내용의 제목을 추출하는 방법을 택하였다. 다양한 종류의 웹 뉴스 페이지를 분석한 결과 뉴스 사이트에서 제목으로 사용되는 태그는 <title>, <meta name="title">, <H1>, <H2> 등 4가지로 나타났다. 이를 바탕으로 콘텐츠 블

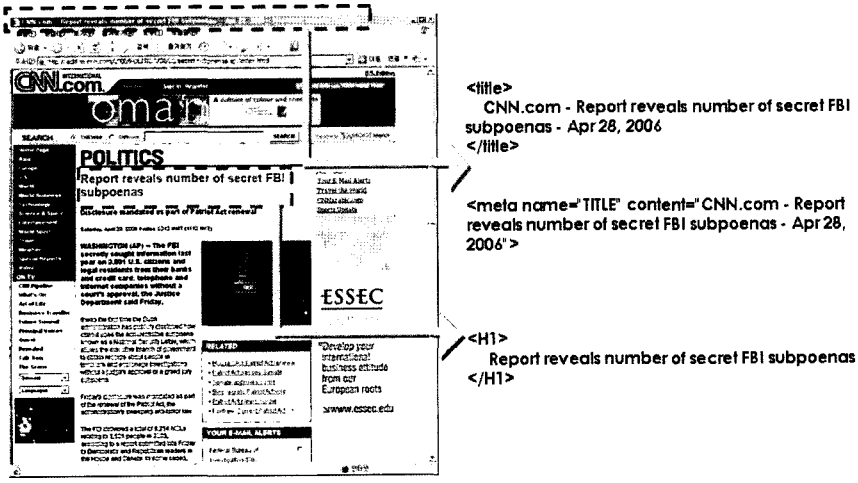

그림 5 제목 추출

록에서 이 4가지 태그만 빼고 나머지 태그들을 제거한 후, 이 4가지 태그와 연관된 내용을 제목으로 사용하였다. 그림 5는 예제 웹페이지에서 제목으로 사용될 수 있는 경우를 보여주고 있다. 이 예에서와 같이 제목이 될 수 있는 태그가 다수 존재하는 경우는 공통으로 포함되는 내용(이 예에서는 "Report reveals number of secret FBI subpoenas")을 제목으로 간주하였다.

3.4 내용 요약

블록 필터링을 통해 분류된 블록 중에서 콘텐츠 블록의 내용을 요약해주는 모듈로서, 요약 정보를 보고 사용자가 원문 전체를 볼 지 결정할 수 있도록 해준다. 내용의 요약은 널리 사용되는 n-gram의 변형된 방법을 이용한 내용 요약후보의 선택을 통해 이루어진다[13]. 본 논문에서 사용된 n-gram의 변형기법의 핵심은 전체 콘텐츠 내용을 토큰(단어) 단위로 나누어서 여러 토큰이 결합된 구(phrase)의 빈도수를 구하여 가장 자주 나타나는 구를 구하는 것이다. 우선 첫 번째 토큰과 두 번째 토큰을 구로 만들고 문서에 몇 번 존재하는지 카운트한다. 2번 이상 존재할 시 다음 토큰을 덧붙여 3개 토큰으로 구성된 새로운 구를 만든다. 이렇게 만들어진 구를 다시 문서에서 몇 번 나타나는지를 구하여 위의 작업을 반복한다. 만약, 2번 이상 존재하지 않을 경우 첫 번째 토큰은 버리고, 새롭게 두 번째 토큰과 세 번째 토큰으로 구를 만들어 위의 작업을 반복한다. 이러한 과정을 반복해서 빈도수가 높은 구를 중요구문으로 선택한다.

그림 6은 중요구문추출 알고리즘의 과정과 의사 코드(pseudo-code)를 나타낸 것이다.

여기서 ❶❷❸은 각각 단어를 나타낸다. 우선, 단어 ❶과 ❷를 구로 가져오고 전체 문서에서 빈도수가 2번 이상 존재하는지 확인한다. ❶❷를 합친 구가 문서에서 2

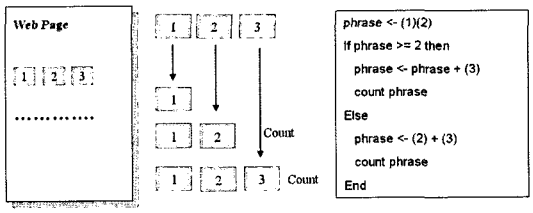

그림 6 중요구문추출 알고리즘

번 이상 존재한다면 단어❸을 합한 구를 만들어 다시 빈도수를 확인한다. 구가 문서에서 2번 이상 존재하지 않는다면, 단어 ❷에 단어 ❸을 합하여 빈도수를 확인한다. 이러한 과정을 반복하여 중요구문을 추출한다.

블록 요약은 추출된 중요구문이 될 가능성이 있는 후보들을 사용한다. 그림 7은 중요구문 추출을 통한 내용 요약 과정을 설명한다. 중요구문 후보들을 사용하여 후보 구들(phrase A,B,C,D,E)이 적어도 하나 존재하는 절(paragraph)을 추출한다. 이것을 요약후보(summary candidate)라 하고 요약 후보들은 예외처리 과정을 거쳐 모바일 디바이스에 나타나는 블록 내용 요약 문장이 된다. 예외처리 과정에서는 문장의 첫 단어가 but, and, so 등과 같은 접속사로 시작하는 문장의 경우 내용의 흐름이 이어지지 않으므로 요약후보에서 제외한다.

그림 8은 웹페이지의 콘텐츠 블록에 대해 중요구문 추출과정과 요약과정을 수행한 다음 모바일 디바이스에 표현한 결과를 보여준다. 웹페이지에서 붉은 테두리로 표시된 부분이 블록 필터링 단계에서 콘텐츠 블록으로 분류된 블록이다. 이 블록에 대해 중요구문 추출 알고리즘을 수행하면 그림의 "Webpage Main"에서와 같이 모바일 디바이스에 붉은 색으로 표시된 제목이 보인다. 이 제목을 선택하면 그림의 "Summary"에서와 같이 블록

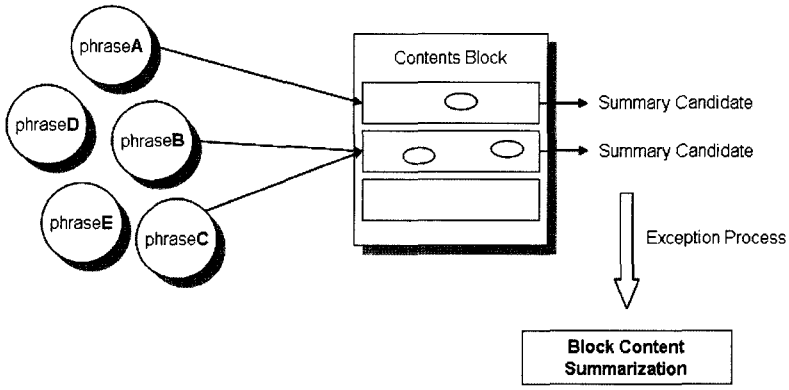

그림 7 중요구문 추출을 통한 내용 요약 과정

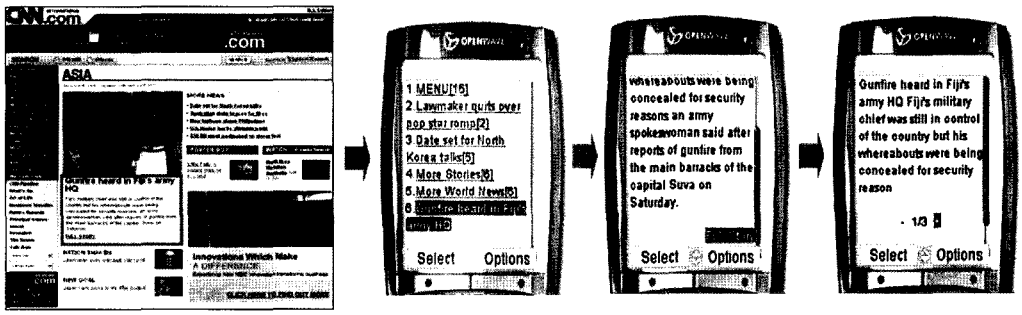

그림 8 모바일 디바이스에 표현된 콘텐츠의 내용요약 결과

의 요약된 내용이 보이고 Full Story라는 링크를 선택하면 그림의 "Full Story"에서와 같이 전체 내용을 볼 수 있다.

3.5 학습을 통한 개인화

블록 필터링, 블록 제목추출, 블록 내용요약의 3단계 과정이 종료되면 모바일 디바이스 적응화가 완료되어 디바이스에 사용자가 보고자 하는 정보가 제공된다. 사용자는 디바이스에서 콘텐츠 블록에서 추출한 제목을 보고 내용을 선택한다. 만일 사용자가 내용의 상세정보(Full Story)를 선택한다면 이것은 사용자가 요약된 내용이 자신이 보고 싶어하는 내용과 연관되어 있다는 것을 암시할 수 있기 때문에 이 정보를 이용하여 사용자 선호주제를 학습할 수 있다. 학습의 과정은 간단하다. 우선 사용자가 상세정보를 선택한 블록 내용에서 빈도수가 가장 높은 단어를 추출하여 사용자 관심정보에 추가한다. 이후 정보검색에서 사용되는 벡터모델(Vector Model)을 이용하여 블록에 존재하는 단어들과 학습에 의한 데이타베이스에 저장된 단어들을 벡터상에 표현한다. 데이타베이스에 저장된 사용자의 관심정보는 벡터모델의 질의로 변환되어 블록 내용과의 유사도를 측정하여 유사도가 가장 높은 순으로 블록의 우선순위가 변경된다. 이러한 학습을 통한 개인화 과정이 그림 9에 도

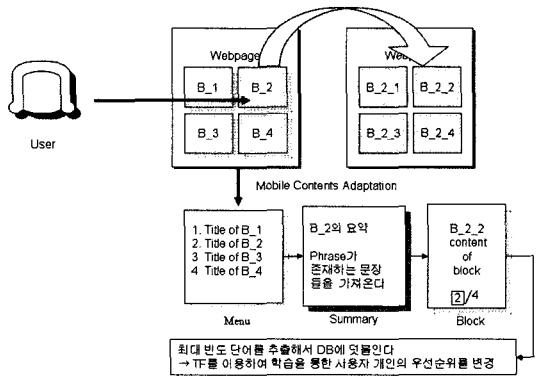

그림 9 학습을 통한 개인화

식화되었다. 이러한 사용자 선호주제 학습의 결과로 사용자가 원하는 정보의 우선순위가 높아져서 해당 주제가 정보 목록에서 상위로 이동되고 원하지 않는 정보는 우선순위가 낮아져서 목록에서 하위로 이동하게 된다. 따라서 사용자에게 차후에 자신이 관심있는 정보를 신속하게 찾아볼 수 있는 편리함을 제공한다.

4. 성능 평가

4.1 실험 환경

표 1 실험 대상 사이트명, 사이트 주소, 페이지수

사이트명	실험에 사용된 URL	실험에 사용된 페이지 수
CNN	http://cnn.com	100
TIMES Magazine	http://www.time.com	100
Los Angeles Times	http://www.latimes.com/	100
New York Times	http://www.nytimes.com/	100
The Times (UK)	http://www.timesonline.co.uk/	100
The Guardian (UK)	http://www.guardian.co.uk/	100
The Independent (UK)	http://www.independent.co.uk/	100

표 1은 실험에 사용된 사이트들과 실험에 사용된 페이지수를 나타낸다. 사이트들간의 비교실험을 위해 유사한 카테고리를 선별하여 사용했다. 실험에 사용된 카테고리들은 Main Page, World, Business, Technology, Science, Entertainment, Weather, Health, Travel이다.

4.2 사이트별 성능비교

본 논문에서는 실험을 위해 7개의 웹사이트를 선별했다. 사이트 내 유사한 카테고리에서 100개의 페이지를 선택하여 블록 필터링이 제대로 되었는지 평가하였다. 실험방법은 System이 웹사이트에서 추출한 카테고리와 실험자가 웹사이트를 보고 분류한 카테고리를 통해 Precision값과 Recall값을 산출하여 F-Measure값을 구한다. 표 2은 사이트별 네비게이션 바(NB Bar), 네비게이션 리스트(NB List), 콘텐츠(Contents)가 제대로 필터링 되었는지 F-Measure값으로 나타내었다.

표 2 사이트별 성능 평가 결과

Site Name	Precision	Recall	F-measure
CNN	84	91	85
TIMES Magazine	95	81	85
Los Angeles Times	81	80	78
New York Times	98	70	79
The Times (UK)	95	72	77
The Guardian (UK)	67	66	60
The Independent (UK)	83	87	84

단위 : %

그림 10, 그림 11은 표 2에서 나타낸 수치값을 카테고리별로 사이트들의 Precision과 Recall을 보여준다.

Precision을 구하는 식은

$$Precision = \frac{System이\ 추출한\ 블럭\ 중\ 실험자가\ 해당\ 카테고리에\ 적당하다고\ 판단되는\ 블럭의\ 수}{System이\ 추출한\ 블럭의\ 수}$$

이고, Recall을 구하는 식은

$$Recall = \frac{System이\ 추출한\ 블럭\ 중\ 실험자가\ 해당\ 카테고리에\ 적당하다고\ 판단되는\ 블럭의\ 수}{실험자가\ 해당\ 카테고리에\ 적당하다고\ 판단한\ 블럭의\ 수}$$

이다.

그림 11에서 LA TIMES의 NB Bar와 GUARDIAN와 INDEPENDENT의 Contents가 다른 사이트들의 카테고리 Precision값보다 낮음을 알 수 있다. 이유는

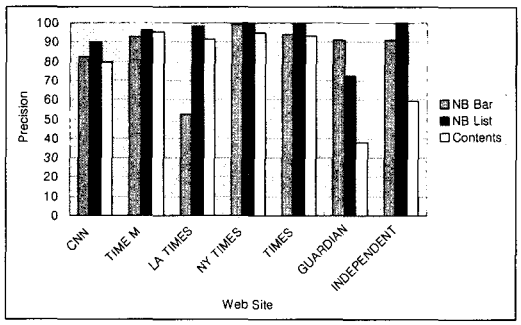

그림 10 사이트별 성능평가 결과 (Precision)

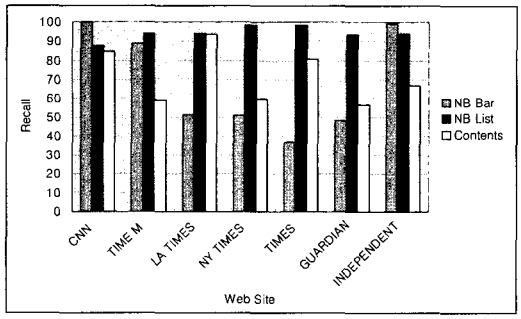

그림 11 사이트별 성능평가 결과 (Recall)

VIPS가 블록 세그멘테이션을 수행할 때 블록을 제대로 나누지 못했기 때문이다. VIPS는 PDoC(Permitted Degree of Coherence) 값을 설정하여 블록 세그멘테이션에 이용한다. PDoC값은 사이트마다 따로 적용할 수 없으므로 PDoC를 8로 설정했다. 이 값에 의해 사이트에 따라 블록이 작게 나뉘질 수도 있고 크게 나눠질 수도 있다. LA TIMES는 PDoC값의 영향으로 블록이 작게 나눠져서 네비게이션바로 표현되지 못하고 하나의 링크들이 각각 블록을 형성하게 되었다. 또 VIPS는 네비게이션 바의 링크들 사이에 선이 존재하게 되면 다른 블록으로 분리하는 단점을 가지고 있다. GUARDIAN와 INDEPENDENT는 PDoC값의 영향으로 블록이 크게 나눠져서 Contents를 대표하는 문장들 뿐만 아니라 이외의 정보도 포함했다. Contents 블록에서 링크가 자바스크립트로 표현되어 있으면 모바일에서는 표현할 수

없으므로 버려지게 된다. 그러므로 실험자는 콘텐츠 블록으로 선택했지만 System은 해당 블록을 제거하므로 Contents블록으로 추출하지 못한다. PDoC의 영향으로 2개의 사이트에서 NB List와 Contents 블록의 Precision은 다른 사이트에 비해 낮았다.

CNN이 NB List 블럭에서 Precision이 낮았다. 실험자는 웹사이트에서 광고를 나타내는 블록임에도 불구하고 리스트 형식으로 표현되면 내용을 포함할 가능성을 가진 링크들의 블록인 NB List로 선택하게 했다. 그러나 System은 다른 도메인을 가지고 있으므로 해당 블록을 제거한다. 그러므로, CNN이 다른 사이트보다 NB Lsit에서 낮은 성능을 보였다.

실험한 사이트들의 평균적인 F-Measure은 78%로 높은 성능을 보이고 있다.

4.4 개인화 평가

본 논문에서는 사용자가 원하는 정보가 우선적으로 모바일 디바이스에 보여주는 개인화가 적용된 콘텐츠 적응화를 제안한다. 개인화가 적용되는 시점은 사용자가 콘텐츠 정보의 상세 내용을 보고자 Full Story의 링크를 선택할 때 시작한다. 내용의 빈도수가 가장 높은 단어를 추출해서 데이타베이스에 저장한다. 저장된 단어들을 쿼리로 사용하여 다른 문서들의 유사성을 비교하여 사용자가 관심을 가지고 있는 내용의 순위를 상위로 바꿔준다.

그림 12는 위에서 설명한 모바일 디바이스에서의 개인화를 실험한 결과이다.

실험자가 모바일 디바이스에서 관심있는 문서를 랜덤하게 선택했을 때 사용자가 관심을 가지는 문서들이 리스트의 상위로 정렬되면 논문에서 제안하는 개인화의 정확도가 어느 정도인지 알 수 있다. 사용자가 콘텐츠의 상세내용을 선택할 때마다 쿼리가 확장되므로 쿼리에 따른 개인화 만족도로 표현된다. 쿼리가 적을 때는 56%의 성능을 나태내고 있고 어느 정도의 쿼리가 모이게 되면 정확도가 97%로 유지됨을 그림 12의 그래프를 통해 알 수 있다.

사용자의 개인화 만족도를 수식으로 표현하면,

$$\text{사용자의 개인화 만족도} = \frac{\text{System이 Query에 적합하다고 추출한 블록에서 사용자가 적합하다고 판단한 블록의 수}}{\text{사용자가 관심을 가지는 블록의 수}}$$

이다.

5. 결론 및 향후 연구과제

본 논문에서 제안하는 웹 콘텐츠 적응화 방법은 데스크탑을 통해서만 볼 수 있었던 방대한 웹 콘텐츠를 사용자의 개인화에 맞게 원하는 정보들이 우선적으로 정렬되어 사용자의 모바일 디바이스에 보여지게 한다.

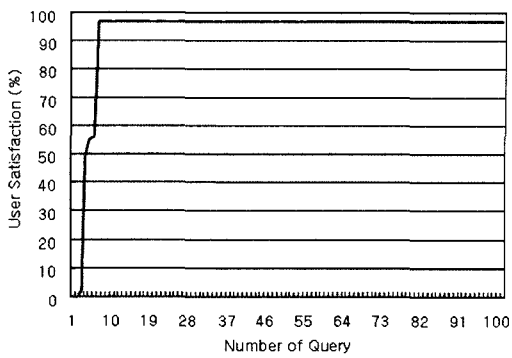

그림 12 사용자의 개인화 만족도

개인화를 통한 웹콘텐츠 적응화 과정은 블록 필터링, 제목추출, 내용 요약, 학습의 4단계로 나눠진다. 블록 필터링은 방대한 웹페이지 정보를 모두 모바일 디바이스에 표현할 수 없으므로 VIPS를 통해 웹페이지를 블록으로 나누고 나눠진 블록에서 광고, 카피라이트와 같은 필요없는 블록은 제거한다. 필요한 블록은 네비게이션 바, 네비게이션 리스트, 콘텐츠의 3개의 카테고리로 나눈다. 콘텐츠 블록에 대해서 제목 추출과 내용을 요약하는 과정이 진행되고 네비게이션 바와 네비게이션 리스트 블록에 대해서는 적응화의 단계를 순차적으로 다시 진행한다. 학습은 콘텐츠에서 요약정보를 보고 난 후 사용자가 상세한 정보를 보기위해 내용을 선택할 때부터 진행되며 사용자 적응화 과정이 시작된다. 학습을 통해 웹페이지에서 사용자가 원하는 내용이 가장 높은 우선순위를 가지며 이를 통해 모바일 디바이스에서 사용자가 원하는 정보는 리스트의 상위에 나타나고 원하지 않는 정보는 리스트의 하위에 존재하게 된다.

위에서 설명한 적응화 과정을 온라인 뉴스 사이트를 통해 실험하고 학습을 통한 개인화 방법이 사용자에게 어느 정도의 만족도를 주는지 실험했다. 첫 번째 실험에서 VIPS를 통해 블록이 잘못 나눠진 3개의 사이트를 제외하고는 78%라는 높은 성능을 보여 주었다. 두 번째 실험에서는 데이타베이스에 질의가 확장됨에 따라 개인화의 정확도가 97%까지 상승되었다.

이러한 성능에도 불구하고 적응화의 정확도를 더 높일 수 있는 기법의 연구가 필요하다. 가장 우선적으로 개선되어야 할 사항으로는 VIPS보다 블록 분류 성능이 좋은 웹 페이지 세그멘테이션 알고리즘을 개발하는 것이다. 이것은 앞에서 기술한 성능저하의 주요 원인이 블록 분류의 문제점에서 기인한 것이라는 것에서 알 수 있다. 본 연구에서는 웹페이지의 태그정보를 이용한 새로운 세그멘테이션 방법을 연구하고 있으며 이 알고리즘이 구현되어 블록 필터링에 사용된다면 지금보다 더 좋은 성능을 보이리라 예상된다. 또한 뉴스 사이트나 온

라인 서점 외에 더 일반적인 웹 사이트에 대한 실험이 진행중이고, 이를 통해 좀 더 범용적이고 확장성이 좋은 모바일 웹 콘텐츠 적응화 시스템을 개발할 수 있을 것이라고 판단된다.

참 고 문 헌

[1] I. Mohomed, J. C. Cai, S. Chavoshi, E. d. Lara, Context-Aware Interactive Content adaptation, 4th International Conference on Mobile Systems, Applications, and Services (MobiSys), Uppsala, Sweden, June 2006.

[2] A. Blekas, J. Garofalakis, V. Stefanis, Use of RSS feeds for Content Adaptation in Mobile Web Browsing, Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A), 2006.

[3] H. Lam, P. Baudisch, Summary Thumbnails: Readable Overviews for Small Screen Web Browsers, Proceedings of the SIGCHI conference on Human factors in computing systems, 2005.

[4] TeliaSonera Finland MediaLab.: Web Content Adaptation - White Paper, http://www.medialab.sonera.fi, 2004.

[5] T. Laakko and T. Hiltunen, Adapting Web Content to Mobile User Agents, IEEE Internet Computing Vol.9, No.2, pp. 46-53, 2005.

[6] A. Pashtan, S. Kollipara, and M. Pearce, Adapting Content For Wireless Web Services, IEEE Internet Computing, Vol.7, No.5, pp. 79-85, 2003.

[7] D. Cai, S. Yu, J. Wen, and W. Ma, VIPS: A Vision-based Page Segmentation Algorithm, Microsoft Technical Report, MSR-TR-2003-79, 2003.

[8] B. Liu, C. Chin, and H. Ng, Mining Topic-Specific Concepts and Definitions On the Web, Proc. 12th Intl. Conf. on World Wide Web, pp. 251-260, 2003.

[9] Y. Hu, G. Xin, R. Song, G. Hu, S. Shi, Y. Cao, and H. Li, Title Extraction from Bodies of HTML Documents and its Application to Web Page Retrieval, Proc. 28th ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 250-257, 2005.

[10] R. Song, H. Liu, J. Wen, and W. Ma, Learning Block Importance Models for Web Pages, Proc. 13th Intl. Conf. on World Wide Web, pp. 203-211, 2004.

[11] J. Chen, B. Zhou, J. Shi, H. Zhang, and Q. Wu, Function-based Object Model Towards Website Adaptation, Proc. 10th Intl. Conf. on World Wide Web, pp. 587-596, 2001.

[12] D. Gokcay and E. Gokcay, Generating Titles for Paragraphs Using Statistically Extracted Keywords and Phrases, Intelligent Systems for the 21st Century, pp. 3174-3179, 1995.

[13] W. Cavnar, J. Trenkle, N-Gram-Based Text Categorization, Proc. SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994.

이 은 실

2002년 동서대학교 컴퓨터공학과(학사) 2006년 한양대학교 대학원 컴퓨터공학과(석사). 2007년~현재 LG전자 MC사업본부 연구원. 관심분야는 인공지능, 데이타 마이닝, 기계학습

강 진 범

2003년~2004년 (주) BnGRotis. 2004년 동명정보대학교 컴퓨터공학과(학사). 2006년 한양대학교 대학원 컴퓨터공학과(석사). 2006년~현재 한양대학교 대학원 컴퓨터공학과 박사 과정. 관심분야는 인공지능, 기계학습, 정보검색, 마이닝

최 중 민

1984년 서울대학교 컴퓨터공학과 졸업(학사). 1986년 서울대학교 대학원 컴퓨터공학과 졸업(석사). 1993년 State University of New York at Buffalo, Computer Science 졸업(박사). 1993년~1995년 한국전자통신연구원(ETRI) 인공지능연구실 선임연구원. 1995년~현재 한양대학교 컴퓨터공학과 교수. 관심분야는 지능형 에이전트, 시맨틱웹, 인공지능, 웹 정보추출