

웹 사이트 탐색 알고리즘 비교분석[†]

(Comparision and Analysis of Algorithm for Web Sites Researching)

김 덕 수* · 권 영 직**
(Deok-Su Kim · Young-Jik Kwon)

요 약 무선 PDA·휴대폰을 통해 웹을 탐색하려는 이용자들은 인터페이스 상의 문제 때문에 어려움을 겪는다. 단지 그래픽을 문자로 바꾸거나 기호체계를 재구성한다고 해서 해결될 문제가 아니다. 심층 연계 구조를 통과하는 데에는 많은 시간이 걸리기 때문이다. 이러한 문제들을 해결하기 위해서 본 논문에서는 실시간의 최단경로를 제공하기 위하여 무선 웹 탐색을 자동적으로 개선시키는 Minimal Path 알고리즘을 제안한다. 본 논문의 결과 Minimal Path 알고리즘은 웹 이용 자들에 대해 지름길을 제공해 주며, 링크의 숫자가 가장 짧았음을 알 수 있었다.

핵심주제어 : 웹 사이트, 탐색 알고리즘

Abstract Visitors who browse the web from wireless PDAs, cell phones are frequently frustrated by interfaces. Simply replacing graphics with text and reformatting tables does not solve this problem, because deep link structures can still require more time. To solve this problem, in the paper we propose an algorithm, Minimal Path Algorithm that automatically improves wireless web navigation by suggesting useful shortcut links in real time. In the result of this paper, Minimal Path algorithm offer the shortcut and the number of shortest links to web users.

KeyWords : Web Site, Minimal Path Algorithm

1. 서 론

1.1 연구의 필요성

경로(Path)[1]란 웹 사이트 방문자가 웹 사이트를 방문한 흔적, 즉 웹 사이트를 방문한 경로를 말한다. 경로는 시간과 공간에 얽매인 웹 사이트 방문자에 의해 요청된 웹 페이지의 연쇄라고 할 수 있다. 연쇄는 웹 사이트가 연결된 서버의 경로를 말한다. 따라서 시

간에 얽매인다는 것은 웹 사이트 방문자가 경로 내에서 연속적으로 요구하는 웹 사이트 요청이 즉시 이루어져야 된다는 것이다. Percowitz와 Etzioni(1997)[2]는 웹 이용자의 접근 패턴을 연구함으로써 구성(organization)과 구현(presentation)을 자동적으로 개선시키는 유용한 웹 사이트 구축을 위해 인공 지능의 세계에 도전했다. 이러한 도전정신으로 그들은 많은 연구 프로젝트를 제안했고 수행했다[10].

이러한 프로젝트의 상당수는, 오늘날의 상당수의 웹과 같이 사이트 방문객이 대형 컬러 디스플레이와 고속의 네트워크 연결을 이용해 탐색하고 있다. 이러한 작업들은 일반적으로 사이트 방문객의 관심이 특정한

† 이 논문은 2002학년도 대구대학교 교내 연구비 지원에 의한 것임.

* 계명문화대학 겸임교수

** 대구대학교 정보통신공학부 교수

목적지를 지향하는 것과는 반대로, 더 많은 페이지의 콘텐츠를 보는 데에 있다. 따라서 이러한 작업들이 강조하는 적합성이란 다음과 같다.

- (1) 실제의 하이퍼텍스트 링크에서 그래픽 요소의 강조
- (2) 많은 페이지와 연계된 인덱스의 자동 생성을 포함한다.

그러나 날로 증가하는 웹 이용자 집단, 즉 핸드폰과 무선 PDA를 통해 콘텐츠를 탐색하는 무선 웹 이용자들은 더 이상 이러한 틀에는 맞지 않는다. 이러한 이동장치는 그들과 짝을 이룬 데스크 탑과 동일한 디스플레이나 대역폭(帶域幅) 적응성을 갖고 있지 않으며, 적합하든 않든 간에 웹 사이트는 데스크 탑 컴퓨터와 이동 장치에 동일한 콘텐츠를 전달하지 않는다. 더욱이 이동 장치를 이용하는 웹 사이트 방문객에 관해 우리가 경험한 바에 따르면, 그들은 많은 페이지를 통해 콘텐츠를 탐색하는 경우가 거의 없었고, 단지 특정한 목적지에만 관심을 갖고 있었으며, 가능한 링크의 숫자가 적은 페이지에 도달하고 싶어 했다. 이러한 관찰이 데스크 탑 이용자의 경우에도 역시 상당 부분 사실이라 하더라도, 이어지는 링크 숫자의 감소는 두 가지의 이유 때문에 이동장치 웹 이용자에게는 특히 아주 매력적이다.[6,8]

첫째, 단지 하나의 페이지에서 특정한 하나의 링크만을 찾을 수 있다면, 이용자는 작은 화면에서 페이지를 따라 스크롤 하는 데 많은 시간을 소비하게 한다.

둘째, 무선 네트워크의 낮은 대역폭과 높은 무응답률 때문에, 하나의 텍스트 페이지에서 하나의 링크만을 연결하는 데에도 수십 초의 시간이 걸릴 수 있다.

따라서 이런 상황에서 4~5개의 페이지를 탐색하게 된다면 쉽사리 몇 분의 시간이 걸릴 것이고, 이용자는 지쳐서 자신의 노력을 포기하게 될 것이다.

Anderson(2001)[1,2,12]에 의하면, 웹 페이지에 도달하는데 필요로 하는 노력(즉 스크롤의 총량과 이어지는 링크의 숫자)과 그 페이지를 봄으로써 얻은 가치를 비교 평가하는 이동통신 이용자를 만족시킬 웹 사이트 적정화의 일반적 틀을 전개하였다. 이 논문에서 이용자가 요구하는 각 페이지에 자동적으로 '지름길 링크'를 부여하는 분석틀에 사용할 수 있는 특정의 적정화가 어떤 것인지 살펴본다. 이동통신 이용자의 행태가 사이트에서 특정한 목적지인 페이지를 봄으로써

수행되는 '정보 수집 과업'에 의해 좌우된다고 가정한다. '지름길 링크'는 다른 방법으로 연결되지 않은 두 개의 페이지를 서로 연결해서 이용자가 가능한 한 신속하게 이러한 목적지에 도달할 수 있게 도와준다. 예를 들어, A→B가 링크를 의미하고 A에서 D로 가는 유일한 경로가 A→B→C→D라면, 지름길 A→D는 이용자가 탐색하는데 드는 노력의 66% 이상을 덜어준다. 중간 페이지에서 스크롤을 피할 수 있다는 것까지 평가한다면 절약되는 노력은 이 이상이다. 그래서 우리는 수준 높은 극소수의 지름길만을 찾는 방법을 제시한다. 따라서 스크롤링 없이 이동통신 장치에 최대한 적합한 지름길만을 생성시키는데 관련연구가 필요하다고 사료된다.

12 연구 방법 및 범위

연구 범위와 방법으로는 첫째 : 이동 매체 이용자들이 실시간 자동적으로 수준 높은 지름길을 찾아낼 수 있도록 해 주는 알고리즘인 Minimal Path 알고리즘을 제시한다. 오프라인에서 Minimal Path는 서버 접근 로그에 기반한 웹 사용 모델을 탐구하고, 이용자의 최종 목적지를 예측하기 위해 실행 시간에 이 모델을 사용한다. Minimal Path 알고리즘은 링크의 유용 가능성과 이용자의 노력 절약을 결합한 지름길의 절약 기대치라는 개념에 근거한다.

둘째 : Naive Bayes 혼합 모델과 Markov 모델 혼합을 포함한 다양한 이용자 모델을 평가하고, Minimal Path에서 그들의 적용 가능성을 논의한다.

셋째 : Markov 모델 혼합이 Minimal Path를 위한 최상의 모델이며, Minimal Path는 이동장치 이용자가 추적해야 할 링크의 숫자, 즉 그들의 정보 탐색에 드는 노력을 실질적으로 감소시킨다는 것을 나타내는 실험적 증거를 제시한다.

본 논문의 구성을 살펴보면 2장은 항해설계 방법론을 제시하고 3장은 예견되는 웹 사용을 위한 모델의 변용 형태를 비교분석하고 4장은 이러한 모델을 사용하는 Minimal Path를 설계 평가한다. 5장은 관련 연구를 논의하고(결과분석) 6장에서 결론을 맺는다.

2. 항해설계 방법론

항해설계 방법론에는 Minimal Path 모델,

Unconditional 모델, Naive Bayes 혼합 모델과 Markov 모델, Positional and Markov/Positional 모델 등이 있는데[10,11,14] 이들에 대해 고찰하였다.

2.1 minimal path 모델

요청된 웹 페이지 p_i 에 소요되는 시간을 $\text{Time}(p_i)$ 라 한다면, 경로 T에는 지나온 페이지와 현재페이지 및 미래의 찾고자하는 페이지가 포함될 수 있다. 그럴 때, 아래와 같이 나타낼 수 있다.

$$\forall i, 0 \leq i < n, p_i \rightarrow p_{i+1} \text{ 인 현재페이지에서 찾고자하는 페이지가 존재한다면}$$

$$\text{Time}(p_i) \leq \text{Time}(p_{i+1}) \leq \text{Time}(p_i) + \text{timeout} \quad (1)$$

인 알고리즘이 존재한다. (여기서 n : 링크되는 숫자, i : 웹 페이지, p_i : 웹 페이지 상에서 현재의 위치이다.)

따라서, 경로의 길이는 연결되어지는 링크의 숫자 n 이다. 경로를 따라가는 이용자의 행태를 관찰하는 적정 웹 사이트의 관점에서 지나온 경로(prefix)가 정해지며, 미래의 경로(suffix)에 대해서 찾고자 할 경우에는 어떤 조건을 줄 필요가 있다.

Minimal Path 모델은 웹 이용자에게 경로가 긴 경로를 단축시키는데 도움을 주는 지름길을 제공하는 것이다. Minimal Path 모델은 이용자가 요청하는 모든 페이지에 지름길 링크를 부여한다. 이론상으로, 지름길은 이용자가 가능한 한 적은 숫자의 링크를 통해 경로의 목적지에 도달할 수 있도록 도와준다는 의미이다. 지름길 선택의 문제를 정확히 나타낸다면 다음과 같다.

예를 들면, 지나온 경로는 각 페이지를 포함하는 것, 즉 (p_0, \dots, p_i) 이며, 지름길의 최대 개수를 m 이라 하면, 지름길의 목록 $p_i \rightarrow q_1, \dots, p_i \rightarrow q_m$ 은 이용자가 p_i 와 목적지 사이에서 추적해야 하는 지름길의 숫자를 최소화시킨다.

지나온 경로 p_i 에서 마지막 페이지는 이용자가 가장 최근에 요청했던 페이지이며, 또한 지름길이 위치

했던 페이지이다. 우리는 하나의 지름길 $p_i \rightarrow q_i$ 에 대해서 이용자가 그 지름길을 따라감으로써 만나지 않았던 것에 대한 지름길의 숫자로 제시한 절약치를 계산할 수 있다. 만일 우리가 전체 경로 $T=(p_0, \dots, p_i, \dots, p_n)$ 의 값을 안다면, 그때 절약되는 링크의 숫자는 수식(2)과 같이 나타낼 수 있다.

$$\begin{cases} j-i-1 & \text{if } q = p_j \text{ for some } i < j \leq n \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

즉, 지름길이 경로를 따라 페이지를 계속 인도할 경우, 절약치를 건너뛴 링크의 숫자로 정의할 수 있다.

여기서 우리는 이용자가 계속 링크, 즉 지름길 링크를 따라가야 하기 때문에 1을 빼야 한다. 만일 지름길이 엉뚱한 곳으로 인도한다면 절약치는 0이며, 최단경로는 이루어지지 않았다고 볼 수 있다.

2.2 Unconditional 모델

가장 단순한 웹 사용 모델은 어떠한 정보에 대해서도 조건을 붙이지 않고 다음 페이지 p_i 를 예측할 수 있다는 것이다. 무조건적 모델은 아래의 수식(3)과 같이 웹 사이트에서의 개별 페이지에 대한 요청된 페이지의 비율을 측정함으로써 얻을 수 있다.

$$P(p_i / q) = \frac{\text{(페이지 } q \text{에 대하여 요청된 개수)}}{\text{(페이지 } q \text{에 대하여 요청된 개수)}} \quad (3)$$

이 모델은 이용자가 단지 현재의 페이지로부터 연결될 때에만 그 페이지를 볼 수 있다고 추정한다. 따라서 Minimal Path는 현재의 페이지로부터 연결되지 않은 페이지의 확률이 제로가 되도록 한다.

2.3 Naive Bayes 혼합 모델

Unconditional 모델은 웹 사이트 상의 모든 경로가 동일하기 때문에 하나의 모델만으로도 이용자의 행태를 정확히 포착할 수 있다고 가정하지만, 많은 사람들은 이러한 가정이 오류라고 생각한다. 즉 서로 다른 웹 사이트 이용자는 서로 다른 경로를 나타낼 것이며,

실사 동일한 웹 사이트 이용자라 하더라도 웹 사이트를 방문할 때마다 각기 다른 경로를 따라갈 수 있다는 것이다. 이러한 문제를 해결하기 위하여 어떤 웹 페이지가 요청될 가능성을 서로 다른 무리(clusters) K 중의 하나(혹은 그 분포)에 속하는 개별 경로로 가정하고 아래 수식(4)과 같이 무리의 동일성 C_k 에 대한 조건화를 통해 페이지의 요청 가능성 q 를 계산할 수 있다.

$$P(\pi = q | \langle p_0, \dots, p_{i-1} \rangle) = \sum_{k=1}^K P(p_i = q | C_k) P(C_k | \langle p_0, \dots, p_{i-1} \rangle) \quad (4)$$

위의 수식(4)에서 서로 다른 모델 K의 확률 추정치 $P(p_i = q | C_k)$ 를 결합한 것이 Naive Bayes 혼합 모델이다. Bayes의 정리(定理)에 의하면, $P(C_k | (p_0, \dots, p_{i-1})) \propto P(C_k)P((p_0, \dots, p_{i-1}) | C_k)$ 이다. 여기서 $P(| C_k)$ 를 계산하기 위해 경로 내에서의 페이지 요청은 주어진 무리와는 독립적이라는 Naive Bayes 가정을 인용한다. 그래서 $P((p_0, \dots, p_{i-1}) | C_k) = \prod_{j=0}^{i-1} P(p_j | C_k)$ 가 존재한다. 결론적으로 Naive Bayes 혼합 모델(AUTOCLASS, Cheeseman외, 1988; 에서 사용된 것과 비슷함)을 통해 우리는 모델의 매개변수 $P(p_i = q | C_k)$ 와, EM 알고리즘을 사용하는 무리 할당 가능성 $P(C_k)$ 를 알 수 있다.

2.4 Markov 모델

Unconditional 모델과 Naive Bayes 혼합 모델은 둘 다 웹 접근, 즉 페이지 경로의 연속적 속성으로부터 얻어지는 핵심 정보 부분을 무시한다. 이에 반해서 이전 페이지에서 다음 페이지로의 추적을 쉽게 할 수 있는 모델이 Markov 모델인데, 이의 수식(5)과 같다.

$$P(p_i = q | p_{i-1}) \quad (5)$$

Markov 모델은 일정한 방식에 따라 다루어지게 되

며, 그 기능대로 움직이게 하는 일련의 데이터에서 페이지 p_{i-1} 에서 p_i 로 이동하는 것을 계산할 수 있다. 또 경로에서 각 페이지가 얼마나 자주 최초의 요청으로 나타나는지를 계산함으로써, 요청된 페이지의 URL을 관련 조작용 데이터의 분량을 증가시키는 URL 아래 경로로 대체시켰다. 이러한 변형의 필요성은 Markov 모델의 경우 더욱 커지는데, 왜냐하면 Markov 모델이 Unconditional 모델보다 2차적으로(제곱값으로) 더욱 커지는 추정 확률값을 가질수록, 결과 ($p_{i-1} \rightarrow p_i$ 로 링크 되는 값)는 더욱 드물기 때문이다.

2.5 Positional and Markov/Positional 모델

수식(6)은 마지막으로 요청되는 웹 페이지에 대한 확률을 조건화 하는 것 외에도 웹 페이지의 이용자의 경로에서 요청한 웹 페이지에 대해 차례로 위치를 조건화할 수 있다. 즉,

$$P(p_i = q | i) \text{ 또는 } P(p_i = q | i, p_{i-1}) \text{ 이다.} \quad (6)$$

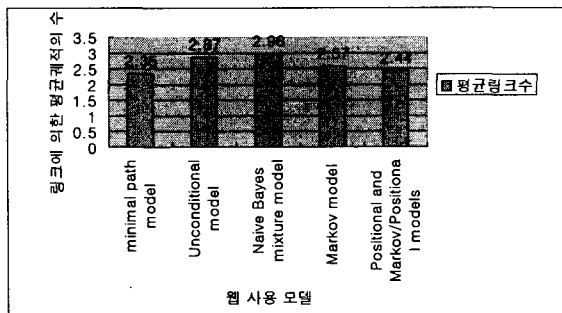
실제로 이 모델은 Unconditional 모델이거나 Markov 모델이거나 간에 경로 내에서 각각의 위치에 대해 개별 모델을 조작(操作)하는 경우에는 비록 우리가 실제적인 목적을 위해 어떤 지점 뒤의 모든 위치를 같은 위치로 취급하더라도 값이 같다.

3. 항해 설계 방법론 비교분석

국내 웹 사이트 중 본 논문에서 정의한 웹 사이트를 유형별로 표본을 선택하여, 경로의 조작군과 1,500개 경로의 시험군을 산출하기 위해 2003년 3월 한달 동안 웹 접근 데이터를 사용하였다. 사이트에서 조작과 시험 기간동안 134,565개의 독자적인 URL의 전체 모집단으로부터 87,561개의 독자적인 페이지가 요청됐다. 그리고 링크 길이가 최소 2개 이상을 가진

경로들만을 선택하였으며, 경로의 깊이 한계를 무한대로 정하여 측정을 하였으며, 경로에 의한 초과시간은 120초로 하였으며, 웹 사용 모델에 대한 링크에 의한 평균 경로의 수는 <그림 1>에서 보여주는 것과 같이 다양한 모델들을 사용할 때 Minimal Path의 효능을 비교하였다.

첫째 칸은 Minimal path 모델에서 이용자가 따라간 링크의 숫자(2.35)를 보여준다. 두 번째 Unconditional model에서는 이용자가 따라간 링크의 숫자(2.87)를 보



<그림 1> 웹 사용 모델에 대한 링크에 의한 평균 경로의 수

였고, 세 번째 Naive Bayes mixture 모델은 사용자가 따라간 링크의 숫자(2.98)를 보였다. 네 번째 Markov 모델은 사용자가 따라간 링크의 숫자(2.57)를 보였다. 마지막으로 Positional and Markov/Positional 모델은 사용자가 따라간 링크의 숫자(2.44)를 나타내었다. 여기서 Markov 모델이 연쇄 정보를 조건화함으로써, 무조건적 모델을 실제로 능가한다는 것을 알 수 있다. 또한, 혼합 모델들이 상응하는 단일 모델보다(예를 들면, Nave Bayes 모델에 대한 2.98 대 무조건적 모델에 대한 2.87) 더 유리한 점을 제공한다는 것을 보여주었다. 실험의 결과 Minimal path 모델은 Unconditional model에 비해 0.52만큼 짧은 링크의 숫자를 보였으며, Naive Bayes mixture 모델에는 무려 0.63만큼의 짧은 링크의 숫자를 기록했다. 그리고 Markov 모델에도 0.22만큼 짧은 링크의 숫자를 기록했으며, Positional and Markov/Positional 모델에도 0.09의 짧은 링크의 숫자를 보여 Minimal path 모델이 가장 우수하다고 할 수 있다.

4. Minimal Path를 이용한 항해설계

아직까지는 일부를 제외한 거의 모든 이동통신 사용자에게 효과적인 콘텐츠를 제공하지 못하고 있는 실정이다. 데스크 탑에서 만큼 이동시간의 제약이 많고, 링크가 많고, 간략하게 표현하는 방법이라든가, 또는 네트워크에 의한 프로세스가 느린 점을 들 수 있다. 이동통신 도구를 사용하는 사람들은 아직까지 이와 같은 이유로 인하여 많은 애로점을 가지고 있는 실정이다. 이에 Minimal path로 인하여 접근하고자 하는 사이트에 접근이 더욱 더 쉽게 이루어질 수 있다.

인터넷에서 요구하는 것은 웹 사이트의 공간을 통해서 적절하게 웹을 탐구하는 것이다. 이런 탐구의 최초 단계는 수정하지 않는 원래의 웹 사이트라는 것이다. 그 단계는 많은 수의 적응기능에 의해 전송이 된다. 이것은 페이지를 만들 수 있고 링크도 할 수 있고, 첨가 또는 삭제할 수 있다. 웹 사이트는 최근에 방문한 실효성으로 측정될 수 있다. 탐색이라는 것은 보다 나은 상태의 웹 사이트가 발견되지 않을 때까지 계속 이루어 질 수 있다. 이와 같은 기능을 지원해 줄 수 있는 알고리즘이 Minimal Path인데, 우선 간단한 정의에 대해서 살펴보면 아래와 같다.

$$\begin{aligned}
 S &= \{W_0, W_1, \dots\} \\
 W &= (\{P_0, \dots, P_n\}, \{L_0, \dots, L_m\}) \\
 L_k &= (P_s, P_d, a) \\
 P_i &= c_i \\
 c_i &= \langle c_{i1}, \dots, c_{ik} \rangle, B \\
 c_j &= \text{text}
 \end{aligned}$$

여기서, 한 화면 전체를 W로 표시하였다. 보충된 시스템이 한번에 하나의 페이지만을 개별화할 수 있도록 전체 웹 사이트를 그 사이트에 있는 어디서든지 만들어 질 수 있도록 전체 웹사이트를 모델화 하고 있다. Po에서 Pn은 기존에 탐색한 사이트이다. Lo부터 Lm은 링크의 개수이다. Ps는 원래 페이지이고, Pd는 목표 페이지이다. a는 연결텍스트이다. 각각 페이지 Pi는 Ci로 모델화 할 수 있다. 이런 웹 콘텐츠는 순서화가 되어 있고 HTML 문서처럼 웹 사이트의 계

층을 따르고 있다. 콘텐츠 cb는 한쌍이고, b는 행동인 자이다. c가 ci-ck까지 포함한다. c는 텍스트 또는 계속 발생하는 노드중 하나이고, b는 사람이 찾아 볼 수 있는 하나의 액션이다. c가 a노드이면 b는 하이퍼텍스트 링크를 불러준다.

위에서 지름길을 간략하게 정의해 보았다. 아래는 minimal path 에 대하여 <그림 2>와 같이 설계해 보았다.

Minimal_Path_Algorithm (Node S, Node D)

/* S: Source node, D: Destination node */

Begin

Path PathList[MAX_PATH_COUNT],

minWeightPath, P;

unsigned int W, pathCount =0;

get_path_list(PathList, PathCount, S, D) /* Find all path from S to D */

For each path P in PathList;

Begin

W ← *evaluate_weight*(P); /* path의 node 개수와 path상의 페이지 크기를 이용하여 weight 산출 */

Set weight W to path P;

End

minWeightPath ← first path in PathList ;

For each path P from the second path in PathList

Begin

If (P's weight < minWeightPath's weight)

Begin

minWeightPath ← P;

Else

End

Return minWeightPath;

End

Get_path_list(Path PathList[], int *PathCount, Node

V, Node D)

/* V: Source node, D: Destination node */

Begin

Path P;

If (V is equal to D) /*

begin

P ← *get_current_path*(); /* stack의 원소들로 구성되는 Path를 산출. 단 Path 산출후 stack의 내용은 이전 상태로 유지 */

Add path P to PathList; /* Path를 목록에 추가. */

(*PathCount)++;

End

Else

Begin

Visited[V] ← TRUE;

push(V); /* Queue에 v를 입력 */

For each non_visited adjacent node W of V

Begin

Get_path_list(PathList, PathCount, W, D);

pop(); /* Remove W from the path stack.

*/

End

End

End

<그림 2> Minimal Path Algorithm

설계 및 구현한 Minimal Path 알고리즘은 <그림 2>와 같다. 이 알고리즘은 탐색한 경로를 저장할 수 있는 테이블을 생성한다. 생성한 테이블에서 경로이름, 위치, 가중치를 포함한 경로의 출발지(S, source node)와 목적지(destination node)에 이르는 전체 경로를 만든다. 만들어진 테이블에서 경로의 node 개수와 경로상의 페이지 크기를 이용하여 가중치를 산출한다. 첫 번째 테이블 처음 경로리스트에서 각 경로 P를 검색한다. 검색한 이들을 비교해서 P가 더 작으면 경로 P를 저장한다. 그렇지 않으면 다시 임의의 저장장소로 이동한다. 만약 방문한 V로드가 목적지와 같은 것을 못 찾았을 때 stack의 원소들로 구성되는 경로를 산출

한다. 단 경로 산출후 stack의 내용은 이전 상태로 유지해야 한다. 그리고 경로를 테이블(목록)에 추가한다. 만약 찾고자 하는 경로가 이미 있을 경우는 V노드에 Queue에 보관하고 다시 경로길이를 가중치에 계산한다. 가중치가 이미 저장된 가중치보다 작을 때 까지 반복해서 찾는다. 이렇게 하여 가장 작은 값을 찾으면 그것이 가장 빠른 길이 된다.

5. 결과 분석

링크 길이가 최소 2개 이상을 가진 경로들만을 선택하는데, 경로가 1개이면 현재 페이지 이므로 탐색에 의미가 없기 때문이었다. Minimal Path 링크의 깊이 한계를 무한대 단계로 정하여 측정을 하였다. 측정결과 다음과 같은 점을 발견 할 수 있었다.

첫째, 이용자의 경로를 따라 목적지로 인도하는 한 개 이상의 지름길이 제시될 때, 이용자는 경로를 따라 가장 멀리 인도하는 지름길을 선택할 것이라는 점이다(다시 말해, 이용자는 외관상 최상으로 보이는 지름길을 욕심스럽게 선택한다는 것이다). 둘째, 이용자의 경로 내에서 페이지를 인도하는 지름길이 없다면, 이용자는 경로 내의 다음 링크를 따라갈 것이라는 점이다(다시 말해, 이용자가 잘못해서 지름길을 따라가지는 않을 것이라는 점이다). 끝으로, Minimal Path는 이용자가 요청하는 각각의 페이지마다 지름길을 배치하며, 그래서 이용자가 한 가지 경로를 따라 다양한 지름길 링크를 따라갈 수 있도록 해준다는 이점이 있다.

6. 결론

이동통신 웹 장치는 조만간 데스크 탑 브라우저들을 수적으로 압도할 것이며, 사이트들은 이동 통신 사용자들의 요구에 적합한 콘텐츠를 제공할 수 있는 준비를 해야만 한다. 이동통신을 이용하는 사용자들은 탐색 비용이 비싸기 때문에 지름길 링크는 현재의 콘텐츠를 증대시킬 신뢰할만한 적용이다.

이에 본 논문에서는 다음과 같은 결과를 얻었다.

첫째, 개별 웹 이용자의 정보 추적 행태를 목적으로

삼은 지름길 링크를 찾아내는 Minimal Path 알고리즘을 전개했다.

둘째, 몇 개의 웹 사용 모델을 살펴보고 그것이 어떻게 Minimal Path를 수행하는지 평가했다.

셋째, Minimal Path가 유용한 지름길 링크를 찾아낼 수 있다는 실험적 증거를 제시했다. Markov 모델 혼합을 사용하여, Minimal Path는 이동통신 이용자로 하여금 링크 절약 가능치의 40% 이상을 절약할 수 있게 한다.

Minimal Path는 신뢰할만한 일련의 관련 연구를 제안하며, 다양한 방향으로 탐구를 계속하고 있다. Minimal Path가 어떻게 더욱 많은 페이지, 더욱 많은 페이지 링크, 그리고 더욱 많은 트래픽을 가진 더 큰 사이트에 대해서도 측정할 수 있는가를 연구하는 것이다.

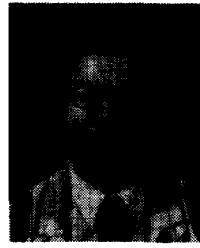
향후 연구 과제로는 지름길 링크에 대한 더욱 간명하고도 설명적인 고정(anchor) 텍스트를 자동적으로 선택할 수 있느냐 하는 것과 웹 개인화에 적용하는 연구가 필요하다.

참 고 문 헌

- [1] Corin R. Anderson, Pedro Domingos, Daniel S. Weld, Adaptive Web Navigation for Wireless Devices, IJCAI-01, 2001
- [2] Corin R. Anderson, Pedro Domingos, Daniel S. Weld, Web Site Personalizers for Mobile Devices, ITWP, 2001
- [3] Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide. Addison- Wesley, Reading, 1999
- [4] De Troyer, O. Audience-driven Web Design, in Conceptual Modeling in the Next Millennium, Rossi, M. CRC Press, USA, 2000.
- [5] B Mirkin. Concept Learning and feature Selection

Based on Square-Error Clustering. Machine learning, 35:25-39, 1999

- [6] O. Zaiane, M. Xin, and J. Han. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In Proc. of IEEE Conference on Advances in Digital Libraries, 1998.
- [7] J. Verner and H. Muller. Management of web site evolution. In Proc. of the International Workshop on Web Site Evolution, Atlanta, GA, USA, October 1999.
- [8] Carroll, J.M. scenario-Based Design:Envisioning Work and Technology in System Development, John Wiley & Sons, 1995.
- [9] J Conallen. Building Web Applications with UML.Addison-Wesley Publishing Company, Reading, MA, 2000.
- [10] Mike Perkowitz. Adaptive Web Site:Cluster Mining and Conceptual Clustering for Index Page Synthesis.PhD thesis, University of Washington, 2001.
- [11] Jacobson, I, Booch, G., and Rumbaugh, J. The Unified Software Development Press, Addison -Wesley, 1999.
- [12] Corin R. Anderson, Pedro Domingos, Daniel S. Weld, Personalizing Web Site for Mobile User, ICM, 2001
- [13] C. Anderson, P. Domingos, and D. Weld. Adaptive web navigation for wireless device. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 2001. Submitted.
- [14] 성기문, 정보제공 웹사이트 평가모형 개발과 검증, 고려대학교석사학위논문, 2000
- [15] 강유화, 서영주, 안성욱저, 무선 이동 네트워크에서의 적응적 자원 할당 방법, 정보과학회논문지, 정보통신 제28호 제4호, 2001, pp477-pp488



김 덕 수 (Deok-Su Kim)

1992년 광주대학교 문헌정보학 (학사)
1995년 대구대학교 산업정보대학원 정보관리전공(공학석사)
2003년 8월 대구대학교 대학원 컴퓨터정보공학과 정보공학전공(공학박사)

1989년~2000년 영남일보 전산부 과장
1998년~현재 계명문화대학 문헌정보과 겸임교수
2000년~현재 (주)나라넷 개발연구 이사
관심분야 : 전자상거래, 이동통신, 정보관리



권 영 직 (Young-Jik Kwon)

1976년 경북대학교 수학과 (이학사)
1980년 영남대학교 경영학과 (경영학석사)
1991년 계명대학교 경영학과 (경영학박사)

1980년~현재 대구대학교 정보통신공학부교수
관심분야 : 소프트웨어공학, 전자상거래