사용자의 이해수준에 따른 효율적인 웹문서 검색

(Efficient Web Document Search based on Users' Understanding Levels)

심 상 희 † 이 수 정 ††
(Sanghee Shim) (Soojung Lee)

요 약 웹 문서 수가 급격히 증가함에 따라 인터넷을 검색할 때마다 발생하는 정보의 과부하 문제가 심각하게 부각되었다. 이러한 문제를 경감시키기 위해 사용자의 선호도에 부합하는 웹 환경을 조성하여 주는 등의 개인화 작업이 주목을 받고 있으나, 대부분의 검색 엔진은 사용자 질의어에만 초점을 두어 응답 결과를 산출하고 있다. 이에 본 논문에서는 사용자의 이해수준에 따른 개인화된 검색 결과를 추출하는 방식에 대해 연구한다. 기존 연구와 차별화된 특징은 사용자 이해 수준을 고려하여 그에 맞는 난이도의 문서들이 우선적으로 검색되게 하는 것이다. 문서에 접근한 사용자들의 이해수준을 바탕으로 문서난이도를 변경시켜 주고, 사용자의 이해수준은 사용자가 접근한 문서 난이도를 바탕으로 주기적으로 변경시켜, 문서 난이도와 사용자 이해수준이 상호 연계되며 변경되도록 하였다. 본 논문의 결과를 적용한 웹 검색 시스템은 다양한 연령층의 웹 사용자들에게 매우 유익한 결과를 가져다 줄 것이다.

키워드 : 웹 검색 개인화, 정보 필터링, 협력 필터링, 사용자 프로필, 추천 시스템

Abstract With the rapid increase in the number of Web documents, the problem of information overload is growing more serious in Internet search. In order to ease the problem, researchers are paying attention to personalization, which creates Web environment fittingly for users' preference, but most of search engines produce results focused on users' queries. Thus, the present study examined the method of producing search results personalized based on a user's understanding level. A characteristic that differentiates this study from previous researches is that it considers users' understanding level and searches documents of difficulty fit for the level first. The difficulty level of a document is adjusted based on the understanding level of users who access the document, and a user's understanding level is updated periodically based on the difficulty of documents accessed by the user. A Web search system based on the results of this study is expected to bring very useful results to Web users of various age groups.

Key words : Web search personalization, information filtering, collaborative filtering, user profile, recommendation system

1. 서 론

웹 문서 수가 급격히 증가함에 따라 인터넷을 검색할 때마다 발생하는 정보의 과부하 문제가 심각하게 부각되었다. 이러한 문제를 경감시키기 위해 사용자의 선호도에 부합하는 웹 환경을 조성하여 주는 개인화 작업이 주목을 받고 있다[1]. 현재 웹 개인화의 가장 성공적인 예는 추천 시스템과 개인화된 웹 검색 시스템이다[2]. 추천 시스템은 주로 고객이 구매하기 원하는 상품을 찾을 수 있도록 도움을 제공하는데 사용된다. 추천시스템과는 달리 개인화된 웹 검색 시스템은 e-commerce 영역에서 상대적으로 낮은 사용도를 보인다. 대부분의 검색 엔진은 사용자가 누구이건 간에 상관 없이 질의어에

· 이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2007-313-D00679)

† 정 회 원 : 경인교육대학원 컴퓨터교육학과
yahoyaho73@hanmail.net
†† 정 회 원 : 경인교육대학원 컴퓨터교육학과 교수
sjlee@gin.ac.kr

논문접수 : 2008년 8월 20일
심사완료 : 2008년 12월 18일

대해 동일한 검색 결과를 제공한다. 따라서 검색 작업에 상당한 시간과 노력을 투자하는데도 불구하고 사용자는 자신이 선호하는 흥미 있는 정보를 얻지 못할 수 있다.

결론적으로 웹상의 정보량이 급격히 증가함에 따라 개인 성향에 부합하는 정보를 찾아주는 개인화 시스템은 매우 중요함에도 불구하고 대부분의 검색 엔진은 사용자 질의어에만 초점을 두어 응답 결과를 산출하고 있다. 최근에 내용 분석이나 사용자 선호도에 근거한 문서 순위를 산출하는 알고리즘을 개발하는 등 앞에서 언급한 문제를 해결하려는 노력이 있어 왔으나 각 사용자의 이해수준에 알맞는 웹 문서들을 산출하는 방법은 아직 개발되지 않았다. 이에 본 논문에서는 사용자의 이해수준에 따른 개인화된 검색 결과를 추출하는 방식에 대해 연구한다. 기존 연구와는 달리, 각 사용자에 따라 차이가 나는 이해 수준을 고려하여 그에 맞는 난이도의 문서들을 우선적으로 검색되게 한다. 이를 위하여 문서 난이도와 사용자 이해수준이 상호 연계되어 변경되게 하고, 그에 따라 사용자 이해수준에 맞는 웹문서가 검색되도록 하고자 한다. 본 논문의 결과를 적용한 웹 검색 시스템은 다양한 연령층의 사용자들에게 매우 유익한 결과를 가져다 줄 것이다.

2. 관련연구

웹 개인화는 특정 사용자의 요구에 맞도록 웹 사이트를 적응시켜 나가는 과정이다. 웹 개인화를 실행하는 두 가지 주된 방법은 협력 필터링[3,4]과 정보 필터링(사용자 프로필)[5,6]이다[7]. 두가지 방법은 모두 사용자가 관심을 보일 만한 항목들을 식별하여 정보 과부하를 경감시키고자 하는 시도이다.

협력 필터링은 여러 다른 사용자들로부터 정보를 획득하여 그들의 의견에 따라 웹사이트를 추천하는 방법이다. 즉, 공통된 흥미를 갖고 있는 사용자들이 선호하는 문서를 제시한다. 과거 수년간 이 방법은 서적, 식품점, 예술과 엔터테인먼트 등 다양한 영역에서 사용되었다[8]. 이 방법의 주요 장점들 중 하나는 추천되는 항목의 내용을 고려하지 않은 채 다른 많은 사람들이 선호한다는 이유 하나만으로 새로운 항목들을 발견할 수 있다는 것이다. 그러나 이 방법의 단점은 새로운 문서에 대해서는 축적된 사용자 선호도 정보가 없기 때문에 새로운 문서를 추천할 수 없다는 것이다. 또 다른 단점으로는 인기 있는 문서를 결정하기 위해 많은 사용자로부터 평가 정보를 필요로 한다는 것이다.

이와 반대로 정보 필터링은 내용 분석을 토대로 개인적인 사용자 흥미도 프로필을 구축하는데 초점을 둔다. 즉, 사용자 요구나 선호도를 반영한 프로필을 필요로 한다. 프로필 구축은 사용자가 직접 입력하거나 사용자의 행위로부터 간접적으로 학습할 수 있다. 예로써, WebMate[9]는 사용자가 흥미를 보이는 문서로부터 간접적이고 자동적으로 사용자의 선호영역을 학습한다. 또한 Persona시스템[10]은 사용자의 검색경로로부터 관심과 비관심 영역을 분류하여 학습한다. 마찬가지로 [11]에서 제안된 시스템은 사용자의 검색이력으로부터 선호 범주를 학습한다. 이러한 방법들은 협력 필터링의 단점을 극복할 수 있는데, 아직 평가되지 않은 새로운 문서에 대해 그 내용을 살펴봄으로써 사용자의 흥미 여부를 예측할 수 있기 때문이다. 그러나 웹문서에 대해 다른 사용자들로부터 어떠한 정보도 필요로 하지 않지만 프로필에 축적된 사용자의 흥미도 외에 새로운 흥미로운 정보를 발견할 수 없는 단점이 있다.

사용자 프로필을 기반으로 하는 정보 필터링과 다른 사용자들로부터 정보를 획득하는 협력 필터링의 각 단점을 극복하기 위한 노력으로 여러 복합 시스템들이 개발되었다. Fab[12]은 정보 필터링 기술을 사용하여 웹 문서 내에서 사용자 흥미 프로필을 보관 및 갱신하였지만 협력 필터링 기법을 동시에 사용하여 유사한 흥미도를 나타내는 프로필을 식별하였다. [13]은 보다 나은 문서를 추천하기 위하여 내용 데이타와 훈련 데이타의 조합을 통하여 Ripper 기계 학습 시스템을 훈련시켰다.

또한 웹 검색 결과를 개선하기 위하여 퍼지 개념을 이용하여 개인화된 문서 검색을 시도한 방법들도 제시되었는데 [14]에서는 사용자 프로필을 퍼지 개념 네트워크으로 구축하고 검색 결과의 문서들과 프로필 네트워크과의 관련성을 문서에 포함된 용어들간의 관계성에 의거하여 계산한 후 각 문서의 순위를 결정하였다. [15]에서는 사용자의 웹 검색 행위를 퍼지 인지 맵(Fuzzy Cognitive Map)으로 표현하여 검색 상태들 간의 관계성을 파악하였다.

3. 사용자의 이해수준에 따른 검색

사용자의 독해능력과 문서 난이도를 객관적으로 정확히 측정할 수 있다면 사용자의 이해수준에 맞는 문서를 정확히 검색해줄 수 있을 것이다. 그러나 사용자의 독해능력을 사용자의 나이, 학력 등의 개인 정보를 통해 객관적으로 정확히 측정하기 어렵고 사용자의 이해수준에 맞는 난이도의 문서를 찾아내는 것 또한 쉽지 않기 때문에 본 논문에서는 사용자가 살펴본 문서들을 통해 사용자의 독해능력을 간접적으로 파악해 그에 맞는 문서를 검색해 주고자 한다. 시스템 구성은 그림 1과 같다.

그림 1과 같이 사용자가 검색엔진을 통해 쿼리를 하게 되면 순위결정모듈은 사용자 이해수준 프로파일과 문서 난이도를 바탕으로 사용자의 이해수준에 가장 가까운 순서로 문서들의 순위를 산출한 후 순위대로 최종

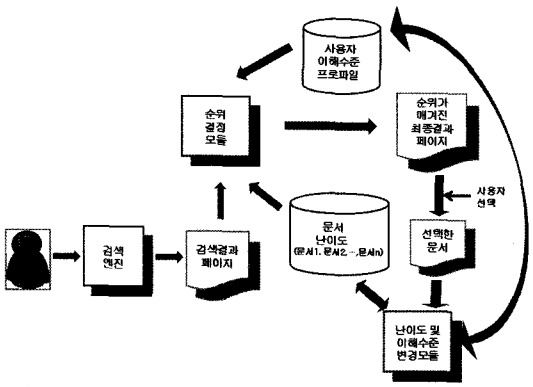

그림 1 사용자의 이해수준에 따른 검색 시스템 설계도

결과 페이지를 출력한다. 최종 결과 문서들에 대해 사용자가 자신이 선호하는 문서들을 선택 조회하면, 난이도 및 이해수준 변경모듈은 선호 문서들의 난이도를 사용자의 이해수준을 반영하여 변경하고, 사용자의 이해수준도 선호 문서들의 난이도를 토대로 변경한다.

그림 1에서 최종 결과 문서들 중 사용자는 선호 문서를 선택하여 조회할 수 있는데, 사용자는 특정 문서에 대한 선호 여부를 직접 또는 간접적으로 나타낸다. WebMate 시스템[9]에서는 직접 방식을 가정하여 웹 개인화를 연구하였으나, 직접 방식은 현실과는 거리가 있으므로, 간접 방식에 대한 선호 여부 판단 방법에 대해 많은 연구가 행해졌다. 사용자가 특정 문서를 단지 열어보았다고 해서 관심 있게 보았다고는 할 수 없으므로, 신뢰할 만한 측정 도구가 필요하며 주요 측정도구로는 문서에 소비한 시간이며 이 밖에도 마우스 클릭 회수, 스크롤바 움직임, 마우스 움직임 등이 있다[16,17]. 그러나 문서 조회 시간이 정확한 측정 도구인가에 대해 부정적인 연구 결과를 찾아 볼 수 있는데, 이는 특정 문서를 열어 놓은 채 방치할 수 있기 때문이다[16]. 이를 보안하기 위하여 [18]의 연구에서는 사용자의 문서 조회 시간의 상한선과 하한선을 두어 범위 내에 있으면, 그 문서를 선호하는 것으로 판단하였다. 이와 같이 효율적인 간접 측정 방식의 개발은 별도의 연구 주제이므로, 본 연구에서는 WebMate 시스템[9]에서와 같이 사용자가 선호하는 문서를 직접 지적하는 방식을 도입하기로 한다. 또한 사용자가 선호하여 조회한 문서는 사용자의 이해수준에 적합한 문서임을 가정한다. 다음 절에서는 난이도 및 이해수준 변경 모듈의 절차에 대해 기술한다.

3.1 문서 난이도 결정

문서 난이도의 결정은 협력 필터링[8]의 방법을 기본으로 사용하여 문서를 선호하는 사용자들의 이해수준을 점진적으로 문서 난이도에 반영하도록 한다. 구체적으로 설명하면 다음과 같다. 문서 난이도의 최소, 최대값을 각각 1과 9라고 하자. 특정 사용자가 임의의 문서를 선호하여 조회할 때마다 선택된 문서의 난이도는 사용자의 이해수준을 반영하여 변경한다. 문서 d의 난이도를 Ld, 임의의 사용자 u의 이해수준을 Lu, 사용자의 이해수준과 문서 난이도의 차를 D(Lu-Ld), 문서 난이도 변경치를 △Ld라고 표기하자. 문서를 검색한 사용자의 이해 수준이 문서 난이도와 비슷한 경우에는 △Ld를 매우 작은 값으로 설정하고, 차이가 크면 점차 △Ld 증감폭을 크게 하고, 각각의 D값 범위(-8≤D≤0 또는 0≤D≤8인 경우)에서 1/2 정도를 지나면 다시 증감폭이 줄어들게 하여 로지스틱 곡선의 S자가 되도록 한다. 이는 적절한 문서 난이도가 정해진 경우라면, 문서 난이도에 맞는 이해수준의 사용자가 접근할 경우가 많으므로 문서 난이도의 변화 폭을 적게 하고 사용자 이해수준과 문서난이도의 차가 클수록 신뢰성이 낮은 값일 수 있으므로 이에 대한 영향을 덜 받게 하기 위한 것이다.

문서 난이도의 변경치 △Ld 곡선이 일정한 값(증가시:0≤△Ld<2, 하락시:-4<△Ld≤0) 내에서 s자 곡선을 그리도록 하는 △Ld 곡선의 함수식은 그림 2에 나타내었다. △Ld의 증가폭은 감소폭보다 1/2 적게 하였는데 이는 문서 난이도가 급격하게 증가하여 낮은 이해 수준의 사용자들이 접근하지 못하게 되는 경우를 적게 하기 위함이고, 대부분의 사용자들은 자기 이해수준보다 낮은 난이도의 문서를 선호하는 경향을 보이는 것을 반영하였기 때문이다. 그리고 문서가 최초 선호되었을 때 문서 난이도는 그 문서를 선호하여 선택한 사용자의 이해수준 값으로 변경되도록 한다.

그림 2의 △Ld 곡선의 함수식은 식 (1)과 같다.

$$\begin{cases} 1.\ D>0,\ f(D)=\dfrac{e^{D}}{1+e^{D}},\ g(D)=f(D-4.0)\text{일 때} \\ \quad \Delta Ld = Max\times g(D)-Max\times g(0) \\ \quad\quad = Max\times\{g(D)-g(0)\} \qquad (1)-1 \\ 2.\ D\le 0,\ f(D)=\dfrac{e^{D}}{1+e^{D}},\ g(D)=f(D+4.0)-1\text{일 때,} \\ \quad \Delta Ld = -\{Min\times g(D)-Min\times g(0)\} \\ \quad\quad = Min\times\{g(0)-g(D)\} \qquad (1)-2 \\ (\text{단, } Max=2.0,\ Min=-4.0) \end{cases} \quad (1)$$

식 (1)에 대한 부연 설명을 하면 다음과 같다. f(x)는 0과 1사이 값을 가진 S자 곡선의 로지스틱 함수로서, 식 (1)-1의 D는 0보다 크므로 f(x)를 x축으로 4.0만큼 축 이동시켜 문서난이도 변경치 △Ld 곡선의 S자가 1사분면에 오도록 하였으며, 사용자 이해수준과 문서난이도의 차 D가 0에 근접할 때는 문서난이도의 변경치 △Ld

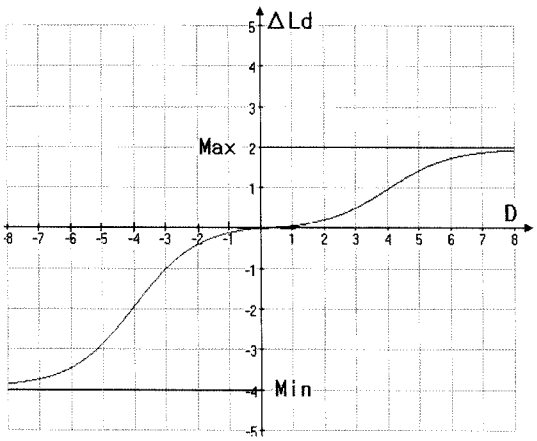

그림 2 문서 난이도 변경치 △Ld의 함수 그래프

또한 0에 근접해야 하므로, g(D)값에서 g(0)를 빼주어 △Ld가 0이 되도록 하였다. 식 (1)-2의 D는 0보다 작으므로 f(x)를 x축으로 -4.0만큼, y축으로 -1.0만큼 축 이동시켜 △Ld 곡선의 S자가 3사분면에 오도록 하였고, D가 0일 때는 △Ld 또한 0이여야 하므로, g(D)값에서 g(0)을 빼주어 △Ld가 0이 되도록 하였다. 또한, Max와 Min은 △Ld의 증감 최대치로서 Max=2.0, Min=-4.0로 정하여 △Ld곡선이 0과 Max와 Min값에 수렴하도록 하였다. 이는 사용자 이해수준과 문서난이도의 차이의 평균값을 기준으로 증감의 Max와 Min값을 정하되, 문서난이도의 증가폭은 감소폭보다 1/2 적게 하여 문서 난이도의 갑작스런 상승을 배제한 것이다.

3.2 사용자 이해수준 결정

그림 1의 난이도 및 이해수준 변경모듈은 문서난이도의 변경 뿐만 아니라, 사용자가 특정 문서에 대해 선호를 표시할 때 이 문서의 난이도를 사용자 이해수준에 반영한다. 즉, 사용자가 방문하는 문서들의 난이도가 높으면 사용자의 이해수준도 함께 높아지고 반대의 경우에는 함께 낮아지게 된다. 사용자의 이해수준은 사용자가 선호하는 문서들의 난이도를 저장해 두었다가 선호문서수(윈도우사이즈) W가 되면 변경시키며, 문서 난이도와 마찬가지로 1부터 9 사이의 실수값을 가진다. 사용자 이해수준의 초기값은 다양한 기준에 의거하여 용도에 따라 부여할 수 있다.

본 논문에서의 사용자 이해수준은 퍼지 멤버쉽 함수를 적용하여 계산하는데, 퍼지집합을 표현하기 위해 자주 사용하는 멤버쉽 함수로는 삼각형(triangle), 사다리꼴(trapezoid), 범종형(gaussian)등의 함수가 있다[19]. 본 논문에서는 가장 흔히 사용되는 범종형 함수를 사용하였으며, 그림 3과 같이 세 개의 퍼지집합, L, M, H와 그에 해당하는 퍼지 멤버쉽 함수 f_L, f_M, f_H를 결정하였다. C_L, C_M, C_H는 각 퍼지집합의 중앙값이다.

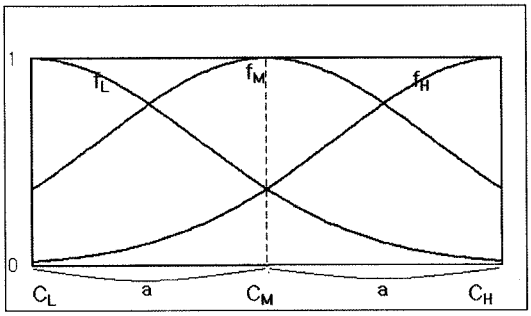

그림 3 사용자 이해수준의 변경을 위한 3개의 퍼지집합과 그에 대응하는 퍼지멤버쉽 함수

사용자 이해수준 Lu는 윈도우사이즈 W 동안 선호하였던 문서들의 난이도에 따라 결정되므로, 사용자가 클릭한 문서들의 난이도와의 차이 평균값과 Lu와의 합, 즉, $x=\sum_{i=1}^{W}(Ld_i-Lu)/W+Lu$를 기준으로, x값에 대응하는 각 퍼지멤버쉽 함수값을 계산한다. 이 때, 계산된 각 함수값은 해당되는 퍼지집합의 중앙값, 즉, C_L, C_M, C_H의 새로운 사용자 이해수준에의 공헌 정도를 나타낸다. 새로운 사용자 이해수준을 구하는 함수식은 식 (2)와 같다. 현재의 사용자 이해수준값을 변경하여야 하므로, C_M을 Lu로 설정하여 계산한다.

$$\begin{cases} C_L = C_M - a,\ C_H = C_M + a & (2)-1 \\ f_L = e^{-(\frac{x-C_L}{a})^2},\ f_M = e^{-(\frac{x-C_M}{a})^2},\ f_H = e^{-(\frac{x-C_H}{a})^2} & (2)-2 \\ Lu1 = C_L \times f_L + C_M \times f_M + C_H \times f_H & (2)-3 \\ Lu2 = \dfrac{Lu1}{f_L + f_M + f_H} & (2)-4 \end{cases} \quad (2)$$

C_M은 현재 사용자의 이해수준이므로 C_M의 범위는 1~9까지이고 범종형의 퍼지함수가 되기 위해서는 C_M으로부터 똑같은 거리로 떨어진 C_L과 C_H가 필요하며 이를 위해 a=4로 정한다. 식 (2)-4는 식 (2)-3에서 계산된 값을 normalization한 것이다. 최종적으로, 사용자의 이해수준이 큰 폭으로 급등락하지 않도록 하기 위해, 변경되기 전의 현재 사용자 이해수준 값을 가중치 β의 비율로 반영하게 한다. 즉, 사용자의 새로운 이해수준을 New_Lu, 가중치를 β, 현재의 사용자 이해수준을 Lu이라고 하면 New_Lu를 구하는 공식은 식 (3)과 같다.

$$New\_Lu=\beta\times Lu+(1-\beta)\times Lu2 \quad (3)$$

4.2절에서 가중치 β값은 0.1에서 0.5까지, 사용자 이해

수준 변경을 위한 윈도우사이즈 W값은 1에서 10까지 다양하게 변화시켜 최적의 β와 W값을 찾는 실험 결과에 대해 기술하였다.

3.3 사용자 이해수준에 따른 문서 순위

그림 1의 순위결정모듈은 사용자의 이해수준에 따른 검색 결과에서 사용자 이해수준과 거의 부합하는 난이도의 문서들을 가장 높은 순위로 하고 차이가 많이 날수록 순위가 떨어뜨린다. 그러나 사용자 이해수준과 문서 난이도의 차이(3.1절의 |D| 값)가 같은 문서의 경우는 어느 문서에 높은 순위를 줄 것인지 결정해야 한다. 이 경우에는 일반적인 사용자의 경향을 반영하여, 사용자 이해수준보다 쉬운 난이도의 문서에 높은 우선순위를 부여하였다.

4. 실험을 통한 성능 평가

4.1 실험 배경

본 논문의 웹실험데이타는 naver 검색엔진을 이용하여 naver검색창에『책 검색-> 네이버책사이트->분야별 책찾기->순수과학->순수과학 베스트셀러 [2008년 1월 1주] 100위(1월1주)』를 바탕으로 총 100개의 문서를 선정하였다. 실험대상은 인터넷을 이용하여 웹 검색이 가능한 초등 5학년 이상부터 50대까지 선정하였으며 실험대상자로 하여금 웹실험데이타 100개의 문서 중에서 자신의 이해수준에 적합하다고 생각하는 웹문서를 모두 고르도록 하였다. 회수된 실험데이타 중 신뢰성이 있는 데이타를 선별하고 그 중 연령대가 고루 분포되도록 선택하여 총 56명의 대상자를 선정하였다.

본 논문의 실험 대상자 이해수준의 초기값은 2부터 8까지로 부여하였으며 최고 및 최저 난이도에 해당되는 1과 9값은 부여하지 않았다. 이는 초등학교 중학년 이하에서는 본 연구의 실험을 이해하고 실험에 참가하는데 어려움이 있고 또한 수집된 데이타의 신뢰성이 떨어질 우려가 있기 때문이다. 따라서 본 논문의 실험은 초등학생 5, 6학년을 대상으로 사용자 이해수준의 초기값을 2로 시작하여 연령대별로 40세까지는 점점 1씩 증가하다 40세 초반부터는 다시 1씩 감한다. 이는 사회적으로 가장 왕성하게 활동하는 시기인 36세이상 40세까지가 경험과 지식이 풍부하다고 판단하여 본 연구 실험 집단의 가장 높은 값인 8을 부여한 것이다. 연령대에 따른 사용자의 이해수준 초기값은 표 1과 같다.

앞에서 언급한 대로 본 논문은 이해수준에 기반하여 개인화된 웹 검색을 실시하는 최초의 연구이므로, 선행연구에서 적절한 비교 대상 시스템을 찾기 어렵다. 따라서 본 논문의 시스템('M'이라 표기)의 성능을 배치 시스템('A'라고 표기)과 네이버 검색엔진 시스템('O'라고 표기)의 성능과 비교하였다. A 시스템이란 실험데이타를

표 1 연령대에 따른 사용자의 이해수준 초기값

사용자의 연령대	사용자 이해수준의 초기값	사용자의 연령대	사용자 이해수준의 초기값
초등학생 5, 6학년	2	31세이상~35세이하	7
중학생	3	36세이상~40세이하	8
고등학생	4	41세이상~45세이하	7
20세이상~24세이하	5	46세이상~50세이하	6
25세이상~30세이하	6	51세이상	5

바탕으로 M 시스템과 같이 문서난이도와 사용자 이해수준을 점진적으로 변경한 것이 아니라, 일괄적으로 처리한 것이다. 즉, 특정 문서의 난이도는 그 문서를 선호한 사용자들의 이해수준 초기값들의 평균으로 결정하였고, 이와 같이 결정된 문서 난이도를 기초로 하여 사용자 이해수준은 사용자가 선호한 문서들의 난이도의 평균값으로 하였다.

평가 기준으로는 기존 연구에서 흔히 사용하는 Precision, Recall을 도입하고, 여기에 추가적으로 Rank값을 새로이 정의하여 사용한다. 성능 측정에 사용할 문서의 범위를 D_N이라 표기하고, 각 시스템에서 산출한 1위부터 D_N 순위까지의 문서들을 기준으로 평가한다. 구체적으로 성능 평가 기준의 계산 방식을 설명하면 다음과 같다. 첫째, Precision값이란 검색 결과 D_N 순위내 문서들 중 사용자가 선호한 문서수의 비율을 나타낸 것이다. M 시스템을 예로 들면, 사용자 이해수준이 5, D_N=10이라면 총 100개의 문서들을 사용자 이해수준인 5에 가까운 순서대로 정렬된 결과 문서들 중, 사용자가 선호한 문서가 10위 안에 3개 있다면 Precision값은 3/10이 되는 것이다. 결과적으로 사용자가 D_N 순위 안에 포함된 문서를 많이 선택하였다면 사용자의 이해수준에 맞는 문서를 잘 검색해준 것이므로 Precision값은 커야 성능이 좋은 것이다. M 시스템의 Precision값을 P_M, A 시스템의 Precision값을 P_A, O 시스템의 Precision값을 P_O로 표기하여 비교하였다.

둘째, Recall값이란 Precision값과 유사한 개념으로 산출 과정이 거의 같으나 단지 나누는 수(제수)를 D_N이 아닌 사용자가 선호한 총문서수로 나눈다. 즉, D_N 순위 안에 있는 문서들 중 사용자가 선호한 문서수가 총 선호 문서수 중 얼마만큼의 비율로 나타나는지를 보기 위한 값이다. 예를 들어, D_N=10, 순위 10위 내 사용자가 선호한 문서수가 3, 사용자가 선호한 총문서수가 15라면 Precision=3/10=0.3이 되나 Recall값은 3/15=0.2가 된다. 그러므로 Recall값을 계산하는 식의 분모는 사용자의 총 선호문서수로 항상 같고 분자는 D_N 값에 따라 달라지므로 D_N이 클수록 Recall값은 커지게 된다. 본 논문의 M 시스템의 Recall값을 RC_M, A 시스템의 Recall값을

RC_A, O 시스템의 Recall값을 RCo로 표기하여 비교하였다.

셋째, Rank값이란 사용자가 선택한 문서들이 D_N 순위 안에서 얼마나 높은 순위를 차지하는지를 알아보기 위한 것이다. D_N 순위 내 문서들 중 사용자가 선호한 문서들의 순위값을 합한 후, 1부터 선호문서수까지의 합으로 나누어 비율로 나타낸다. 예를 들어, D_N=10이고 순위 10위 내 선호 문서들의 순위가 2, 5, 7이라면 Rank=(2+5+7)/(1+2+3)=2.33이 된다. M 시스템의 Rank값을 RK_M, A 시스템의 Rank값을 RK_A, O 시스템의 Rank값을 RK_O로 표기하여 비교하였다. Rank값을 구하는 식의 분모와 분자가 같으면 최적의 검색 결과이므로 Rank값은 1에 가까울수록 우수하다.

4.2 파라미터값 결정

본 논문 시스템의 성능 향상을 위한 파라미터 값으로 가중치 β값과 윈도우 사이즈 W의 최적치를 실험을 통해 결정하였다. β값이란 사용자의 이해수준을 변경시킬 때 현재의 사용자 이해수준을 얼마나 반영해줄 것인가를 결정하는 가중치이다. 새로운 Lu값을 더 많게 또는 최소한 같은 비중으로 반영시키기 위하여 β값은 0.1에서 0.5까지 변화시켜 적정값을 찾았다. W값이란 사용자 이해수준을 변경시킬 주기를 의미하는 것으로 사용자가 선호하여 클릭한 문서수를 말한다. 웹실험데이타 문서수가 100개이고 웹실험데이타 사용자는 평균 20개 전후의 문서를 선택하였으므로 W값은 1에서 10까지 변화시켜 최소 1회 이상 변경될 수 있도록 하였다. 최적의 파라미터 값을 찾기 위해 사용자 수는 총웹실험데이타 사용자 수인 56으로 하고 D_N 값은 20으로 정한다. D_N=20으로 정한 이유는 사용자 56명이 각자의 이해수준에 맞는 문서를 100개 중 평균적으로 22.1개 선택했기 때문이며 실험 결과는 표 2, 3, 4와 같다.

표 2 W=1일 때의 β값 변화에 따른 실험 결과

β	P_M	RC_M	RK_M
0.1	0.306	0.297	3.226
0.2	0.333	0.324	2.915
0.3	0.342	0.329	3.066
0.4	0.354	0.341	2.774
0.5	0.367	0.355	2.579

표 3 W=5일 때의 β값 변화에 따른 실험 결과

β	P_M	RC_M	RK_M
0.1	0.422	0.409	2.757
0.2	0.429	0.417	2.483
0.3	0.442	0.429	2.373
0.4	0.451	0.438	2.237
0.5	0.471	0.453	2.212

표 4 W=10일 때의 β값 변화에 따른 실험 결과

β	P_M	RC_M	RK_M
0.1	0.444	0.428	2.415
0.2	0.446	0.430	2.227
0.3	0.458	0.438	2.150
0.4	0.456	0.437	2.241
0.5	0.454	0.436	2.473

표 5 β=0.5일 때의 W값 변화에 따른 실험결과

W	P_M	RC_M	RK_M
1	0.367	0.355	2.579
2	0.383	0.374	2.820
3	0.419	0.411	2.802
4	0.440	0.428	2.341
5	0.471	0.453	2.212
6	0.476	0.458	2.148
7	0.479	0.459	2.210
8	0.469	0.450	2.267
9	0.465	0.447	2.265
10	0.454	0.436	2.473

위 표에 제시된 것과 제시하지 않은 추가적인 실험을 종합한 결과에 따르면 윈도우 사이즈 W가 1, 3-7까지일 때는 가중치 β값이 0.5, W가 2일 때는 β값이 0.4, 8-10일 때는 β값이 0.3일 때가 가장 우수한 성능을 나타냈다. 본 논문은 실험자수가 56명으로 적은 인원이므로 현재 사용자 이해수준의 반영 비율을 크게 하고자 최적의 β값을 0.5로 정하였다. 또한 최적의 W값을 결정하기 위해 β값이 0.5일 때의 W값을 변화시켜 실험하였으며, 결과는 표 5와 같이, W값이 7일 때 가장 좋은 성능을 보였다. 따라서 최적의 파라미터값을 β=0.5, W=7로 결정하여 다음 절에서 제시한 대로 실험을 진행하였다.

4.3 성능 평가

4.3.1 D_N 순위에 따른 성능 비교

총사용자 56명의 D_N 순위에 따른 Precision값은 표 6과 같다. 세 시스템 모두 D_N 순위가 커질수록 값이 조금씩 떨어졌고 본 시스템 값의 낙폭이 상대적으로 더 컸다. 이는 시스템 초기에 문서 난이도와 사용자의 이해수준이 결정되어 가는 과정이기 때문인 것으로 파악되며, 어느 정도 시스템이 운영되어 안정된 문서 난이도와 사용자의 이해수준이 결정되면 Precision값이 소폭의 증감을 거듭하는 일정한 패턴을 유지할 것으로 본다. 또한 M 시스템의 Precision값은 A 시스템 결과와는 비슷하며 네이버 검색엔진인 O 시스템보다는 최대 0.213까지 컸다. 이는 본 논문 시스템이 사용자의 이해수준에 더 적절한 문서를 검색해 준다는 것을 의미한다.

총사용자 56명의 D_N 순위에 따른 Recall값은 표 7과 같다. 세 시스템 모두 D_N 순위가 클수록 값이 커졌다.

표 6 D_N 순위에 따른 Precision값 비교

D_N	P_M	P_A	P_O
10	0.516	0.463	0.310
15	0.501	0.444	0.288
20	0.479	0.439	0.285
25	0.446	0.444	0.273
30	0.424	0.429	0.271
35	0.404	0.408	0.266
40	0.381	0.395	0.254

표 7 D_N 순위에 따른 Recall값 비교

D_N	RC_M	RC_A	RC_O
10	0.249	0.221	0.143
15	0.364	0.318	0.201
20	0.459	0.416	0.262
25	0.532	0.525	0.316
30	0.603	0.604	0.373
35	0.665	0.668	0.429
40	0.713	0.733	0.466

표 8 D_N 순위에 따른 Rank값 비교

D_N	RK_M	RK_A	RK_O
10	2.078	2.358	3.122
15	2.298	2.362	4.167
20	2.210	2.401	3.933
25	2.223	2.397	3.915
30	2.244	2.397	3.851
35	2.368	2.436	3.861
40	2.399	2.455	3.907

이는 Recall값을 정의할 때 설명했듯이 D_N 순위 내 문서들 중에서 사용자가 선호한 문서수를 그 사용자의 총 선호문서수로 나누었기 때문이다. 또한 M 시스템의 Recall값은 A 시스템과 비슷하며 O 시스템보다는 최대 0.247까지 컸다. 이는 본 논문 시스템이 사용자의 이해수준에 더 적절한 문서를 검색해 준다는 것을 의미한다.

총사용자 56명의 D_N 순위에 따른 Rank값은 표 8과 같다. M 시스템의 Rank값은 A 시스템 결과와 비슷하며 O 시스템 결과보다는 최대 1.044까지 값이 작았다. Rank값은 1에 가까울수록 사용자의 이해수준에 가까운 것을 검색해준다는 의미이므로, 본 논문 시스템의 성능이 보다 우수함을 알 수 있다.

4.3.2 사용자수에 따른 성능 비교

사용자수의 변화가 본 시스템에 어떤 영향을 미치는지 알아보기 위하여 D_N 순위 20에 대해 성능 비교 실험을 하였고 그 결과는 표 9, 10, 11과 같다.

D_N 순위 20에 대해 사용자수에 따른 본 논문 시스템과 다른 두 시스템의 Precision값을 비교하였더니 표 9

표 9 사용자수에 따른 Precision값 비교

사용자수	P_M	P_A	P_O
10	0.500	0.460	0.330
15	0.483	0.470	0.303
20	0.480	0.468	0.300
25	0.458	0.478	0.304
30	0.462	0.470	0.310
35	0.430	0.447	0.300
40	0.423	0.439	0.304
45	0.417	0.437	0.293
50	0.428	0.447	0.294
55	0.432	0.443	0.287

표 10 사용자수에 따른 Recall값 비교

사용자수	RC_M	RC_A	RCo
10	0.423	0.374	0.277
15	0.430	0.407	0.262
20	0.437	0.415	0.264
25	0.412	0.415	0.260
30	0.416	0.411	0.269
35	0.389	0.393	0.260
40	0.377	0.382	0.264
45	0.395	0.400	0.261
50	0.414	0.419	0.267
55	0.421	0.419	0.263

표 11 사용자수에 따른 Rank값 비교

사용자수	RK_M	RK_A	RK_O
10	1.831	2.143	3.324
15	2.169	2.224	3.561
20	2.201	2.212	3.315
25	2.250	2.161	3.416
30	2.252	2.176	3.246
35	2.341	2.348	3.483
40	2.390	2.423	3.472
45	2.515	2.416	3.872
50	2.475	2.350	3.782
55	2.467	2.380	3.911

와 같이 세 시스템 모두 사용자수가 많을수록 값이 처음보다 조금씩 떨어졌고, M 시스템의 등락폭이 약간 더 컸다. 이는 4.3절 (1)의 Precision값 결과와 마찬가지로 시스템 초기에 문서 난이도와 사용자의 이해수준이 결정되어 가는 과정이기 때문인 것으로 보인다. 또한 M 시스템의 Precision값은 A 시스템 결과와 비슷하고 O 시스템보다는 사용자수에 상관없이, 최대 0.180까지 더 우수한 성능을 보였다.

Recall값을 비교하였더니 표 10과 같이 M 시스템과 A 시스템 값의 등락폭이 상대적으로 컸고 O 시스템은 거의 일정한 값을 유지하였다. 또한 세 시스템 모두 사

용자수에 무관한 성능을 보임을 알 수 있다. M 시스템의 Recall값은 A 시스템과 비슷하며 O 시스템보다는 최대 0.173까지 컸다.

Rank값을 비교한 결과는 표 11과 같다. M 시스템의 Rank값은 A 시스템과 비슷하며 O 시스템보다는 전반적으로 1.3~1.4 정도 작았으며, 최대 1.493의 차이를 보였다. 이는 사용자수에 무관한 성능이며, 본 논문 시스템이 사용자의 이해수준에 적절한 문서들에게 보다 높은 우선순위를 부여한다는 것을 의미한다.

4.3.3 문서난이도와 사용자 이해수준의 최종 수준별 분포

본 논문 시스템의 실험에 사용된 문서들의 난이도의 초기값은 설정하지 않고 그 문서를 선호한 사용자의 이해수준에 따라 변경되도록 하였다. 반면, 사용자 이해수준의 초기값은 4.1절의 표 1과 같은 기준에 의해 표 12와 같이 균등분포가 되도록 하여 실험하였다. 실험 결과, 문서난이도와 사용자 이해수준값의 최종 수준별 분포는 최적의 파라미터값인 β=0.5, W=7, D_N=20일 때, 표 13과 같다. 문서난이도와 사용자 이해수준 값은 2.0 미만과 8.0이상의 값을 가지지 않았으며 이는 실험데이타의 초기 이해수준 값을 2.0에서 8.0까지로 부여하였기 때문이다. 또한 문서난이도와 사용자 이해수준의 수준별 분포값이 중앙에 집중되어 있는 것을 볼 수 있는데 그 이유는 다음과 같다. 문서 난이도가 최저치 2일 경우, 이 문서를 선호하는 사용자들은 2보다 큰 이해수준을 갖는 사용자들이 많을 것이므로, 이 문서의 난이도는 상향 조정될 수밖에 없다. 이와 반대로, 문서 난이도가 최고치일 경우도 마찬가지로 추론해 볼 수 있다. 또한 2~5까지의 난이도를 갖는 문서나 사용자 총수가 5~8까지보다 많은 이유는 3.1절에서 설명한 대로 문서난이도의 증가치를 감소치의 1/2로 하였기 때문이다.

표 12 사용자 이해수준의 초기값 수준별 분포

사용자 이해수준 초기값	2	3	4	5	6	7	8	계
사용자수	8	8	8	8	8	8	8	56

표 13 문서난이도와 사용자 이해수준 값의 최종 수준별 분포

수준분포(B)	문서수	사용자수
1.0≤B<2.0	0	0
2.0≤B<3.0	7	8
3.0≤B<4.0	14	10
4.0≤B<5.0	35	13
5.0≤B<6.0	25	17
6.0≤B<7.0	15	5
7.0≤B<8.0	4	3
8.0≤B<9.0	0	0
계	100	56

5. 결론 및 향후 연구과제

본 논문은 문서 난이도를 문서에 접근한 사용자의 이해수준을 이용하여 변경시켜 주고, 사용자의 이해수준도 사용자가 접근한 문서들의 난이도를 바탕으로 주기적으로 변경시켜, 문서 난이도와 사용자 이해수준이 상호 연계되며 변경되도록 하여, 사용자의 이해수준에 적합한 문서를 우선적으로 검색해주는 개인화된 웹 검색방법을 제안하였다. 결과적으로 검색된 문서들에게 개인화된 순위를 부여해 주는 것이며, 기존의 개인화 검색 시스템에 관련된 연구에서는 사용자의 선호도만을 파악하였으나, 본 논문은 사용자의 이해수준을 고려하여 그에 맞는 난이도의 문서들이 검색되도록 하는 최초의 연구이다. 사용자 이해수준의 파악은 웹 검색 과정 동안 간접적으로 이루어지고 이를 측정하기 위하여 윈도우사이즈를 사용하므로 과거의 기록을 반영하게 된다. 본 논문의 시스템은 문서의 난이도를 고려하였으므로 특히 지식이 미성숙된 사용자들(예: 유아, 초등생)에게 신속하고 유익하게 사용될 수 있다.

향후 연구과제로는 본 논문이 실제적으로 시스템을 구축하여 실험한 것이 아니므로 실험데이타 수가 적고 다양한 환경과 조건을 고려하지 않았으므로, 본 논문의 문서 난이도와 사용자의 이해수준 결정 과정을 바탕으로 한 시스템을 구축하여 실제 환경에서의 보다 철저한 실험이 필요하다.

참 고 문 헌

[1] D. Arotaritei and S. Mitra, "Web mining: a survey in the fuzzy framework," Fuzzy Sets and Systems, Vol.148, 2004. pp. 5-19.

[2] C. Shahabi and Y.-S. Chen, "Web Information Personalization: Challenges and Approaches," 3rd International Workshop on Databases in Networked Information Systems(DNIS). Aizu-Wakamatsu, Japan. pp. 5-15, 2003.

[3] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, "Visualization of navigation patterns on web site using model based clustering," Technical Report MSR-TR-00-18, Microsoft Research, Microsoft Corporation, Redmond, WA, 2000.

[4] C.J. van Rijsbergen. Information Retrieval. Butterworths, 1979.

[5] B. Mobasher, R. Cooley, and J. Srivastava, "Creating adaptive web sites through usage-based clustering of URLs," Proc. IEEE Knowledge and Data Engineering Exchange Workshop, pp. 19-25, 1999.

[6] A. Stefani and C. Strapparava, "Exploiting nlp techniques to build user model for web sites: The use of worldnet in SiteIF project," Proc. 2nd Workshop on Adaptive Systems and User Modeling on the WWW, 1999.
[7] H.-R. Kim and P. K. Chan, "Personalized Search Results with User Interest Hierarchies Learnt from Bookmarks," 7th International Workshop on Knowledge Discovery on the Web, WebKDD 2005.
[8] N. Good, J. Schafer, J. Konstan, J. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations," Conference of the American Association of Artificial Intelligence. pp. 439-446, 1999.
[9] L. Chen and K. Sycara, "WebMate: A Personal Agent for Browsing and Searching," 2nd International Conference on Autonomous Agents(Agents '98). Minneapolis, USA: ACM Press. pp. 132-139, 1998.
[10] F. Tanudjaja and L. Mui, "Persona: A Contextualized and Personalized Web Search," The 35th Annual Hawaii International Conference on System Sciences(HICSS'02). Big Island, Hawaii, 2002.
[11] F. Liu, C. Yu, and W. Meng, "Personalized web search for improving retrieval effectiveness," IEEE Trans. Knowl. Data Eng., Vol.16, No.1, pp. 28-40, 2004.
[12] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," Communications of the ACM, Vol.40, No.3, pp. 66-72, March. 1997.
[13] C. Basu, H. Hirsh, and W.W. Cohen, "Using Social and Content-Based Information in Recommendation," Proceedings of the AAAI-98, 1998.
[14] K.-J. Kim and S.-B. Cho, "Personalized mining of web documents using link structures and fuzzy concept networks," Applied Soft Computing 7, pp. 398-410, 2007.
[15] G. Meghabghab, "Mining user's web searching skills through fuzzy cognitive state map," Proc. Joint 9th IFSA World Congress and 20th NAFIPS Internat. Conf., pp. 429-434, 2001.
[16] K. Jung, "Modeling web user interest with implicit indicators," Master Thesis, Florida Institute of Technology, 2001.
[17] L.A. Granka, T. Joachims, and G. Gay, "Eye-tracking analysis of user behavior in WWW search," In Proc. 27th annual international conference on Research and development in information retrieval, 2004.
[18] H. Kim and P.K. Chan, "Implicit indicator for interesting web pages," International Conference on Web Information Systems and Technologies, pp. 270-277, 2005.
[19] 유동선,이교원, 기초 퍼지 이론, 교우사, 서울, 2001.

심 상 희
1996년 전주교육대학교 졸업(학사). 2008년 경인교육대학교 대학원 컴퓨터교육학과. 관심분야는 컴퓨터 교육, 개인화된 웹 검색

이 수 정
1985년 이화여자대학교 졸업(학사). 1990년 Texas A&M 대학교 컴퓨터공학과 졸업(석사). 1994년 Texas A&M 대학교 컴퓨터공학과 졸업(박사). 1995년~1998년 삼성전자 통신개발실 선임연구원 1998년~현재 경인교육대학교 컴퓨터교육과 교수. 관심분야는 컴퓨터교육, 인간 컴퓨터 상호작용, 라우팅 알고리즘, 신경망, 개인화된 웹 검색