지능정보연구 제16권 제4호 2010년 12월(pp.21~41)

적응형 사용자 프로파일기법과 검색 결과에 대한 실시간 필터링을 이용한 개인화 정보검색 시스템

전호철
한양대학교 컴퓨터공학과
(hcjeon@cse.hanyang.ac.kr)

최중민
한양대학교 컴퓨터공학과
(jmchoi@hanyang.ac.kr)

본 논문은 다양한 사용자의 개인적 검색요구를 충족시키지 못하는 기존 검색시스템의 문제점을 해결하기 위해 사용자의 묵시적 피드백을 이용한 적응형 사용자 기호정보 기반의 개인화 검색을 실현하고, 검색결과에 대한 실시간 필터링을 통해 사용자에게 적합한 검색 결과를 제공하는 시스템을 제안한다. 기존의 검색 시스템들은 검색의도의 불확실성 때문에 사용자의 검색실패율이 높다. 검색 의도의 불확실성은 동일한 사용자가 "java"와 같은 다의어에 대해 동일한 질의어를 사용하더라도 다른 검색 결과를 원할 수 있다는 것이며, 단어의 수가 적을수록 불확실성은 가중될 것이다. 실시간 필터링은 사용자의 도메인 지정여부에 따라 주어진 도메인에 해당하는 웹문서들만 추출하거나, 적절한 도메인을 추론하고 해당하는 웹문서들만 검색 결과로 보여주는 것으로, 일반적인 디렉토리 검색과 유사하지만 모든 웹문서에 대해 이루어진다는 것과 실시간으로 분류된다는 것이 다르다. 실시간 필터링을 개인화에 활용함으로써 검색결과의 수를 줄이고 검색만족도를 개선했다. 본 논문에서 생성한 기호정보파일은 계층적 구조로 이루어지며, 상황정보의 반영이 가능하기 때문에 의도의 불확실성을 해결 할 수 있다. 또한 사용자의 도메인별 웹문서 검색 동작을 효과적으로 추적(track) 할 수 있으며, 사용자의 기호 변화를 적절하게 알아낼 수 있다. 각 사용자 식별을 위해 IP address를 사용했으며, 기호정보파일은 사용자의 검색 행동에 대한 관찰을 기반으로 지속적으로 갱신된다. 또한 사용자의 검색결과에 대한 행동 관찰을 통해, 사용자 기호를 인지하고, 기호정보를 동적으로 반영했으며, 검색결과에 대한 만족도를 측정했다. 기호정보파일과 반영비율은 사용자가 검색을 수행할 때 시스템에 의해 생성되거나 갱신된다. 실험결과 적응형 사용자 기호정보파일과 실시간 필터링을 함께 사용함으로써, 상위 10개의 검색결과 중 평균 4.7개의 결과들에 대해 만족하는 것으로 나타났으며, 이는 구글의 결과에 비해 약 23.2% 향상된 만족도를 나타내었다.

논문접수일 : 2010년 09월 03일 논문수정일 : 2010년 11월 15일 게재확정일 : 2010년 11월 21일 교신저자 : 최중민

1. 서 론

웹 상의 접근 가능한 문서들의 수가 기하급수적으로 증가함에 따라 사용자는 자신이 필요로 하는 정보를 얻기 위해 검색을 시도하고 있지만, 대다수의 사용자는 자신이 원하는 정보를 쉽게 얻기 어렵다. 이것은 구글, Yahoo, MSN같은 기존의 검색엔진들이 각 사용자의 개인 정보, 검색 성향 또는 검색 의도 인지 없이 사용자가 입력한 질의어와 관련된 정보들만 추출하고 제공하기 때문이며, 따라서 기존 검색 엔진에 의한 검색 결과는 다양한 사용자의 요구를 충족시키기 어렵다.

일반적으로, 각 사용자에 특성화된(Personalized) 검색 결과를 제공하기 위해, 사용자 기호정보파일(user preference profile) 기법이 사용되어 왔다(Wang, J. et al., 2006). 이것은 사용자의 기호

도를 검색 결과들과 비교하여 불필요하다고 여겨지는 결과들을 제거한 뒤 사용자에게 제공해 주는 것이다. 사용자 기호도 프로파일 기법은 기호정보 갱신 방법에 따라 정적 프로파일 기법과 동적 프로파일 기법으로 구분 할 수 있다. 정적 프로파일 기법은 사용자 스스로 자신의 기호정보를 선택하거나 지정하기 때문에 매우 정확하다는 장점이 있는 반면에, 부정확한 사용자 기호정보를 포함할 뿐만 아니라, 사용자가 직접 변경을 해야 하는 단점이 있다. 그런가 하면, 동적 프로파일 기법은 사용자의 어떠한 개입 없이 시스템에 의해 사용자의 기호정보를 추론하고 유지하기 때문에 불확실성을 지니는 반면, 그 과정이 자동적으로 이루어지며, 지속적으로 사용자의 기호정보를 갱신하기 때문에 상대적으로 부정확한 사용자 기호정보를 포함할 확률이 낮다는 장점이 있다.

부정확한 사용자 기호정보는 사용자 기호정보의 내용과 현재의 사용자 기호정보가 일치하지 않는 것을 의미하며, 이것은 사용자 기호정보의 가변성과 불확실성에 기인한다. 사용자 기호정보의 가변성은 시간에 따라 사용자의 기호가 바뀌는 것을 의미하며, 이것은 장시간 변하지 않는 장기기호(long-term preference)와 특정 순간 또는 짧은 기간 동안 관심을 갖는 단기기호(short-term preference)로 구분된다. 반면에, 사용자 기호정보의 불확실성은 동일한 질의어더라도 그 의도가 다르기 때문에 동일한 검색 결과 리스트에 대해서도 다른 검색 결과문서를 선택할 수 있는 것을 의미한다.

많은 다양한 분야에서 이러한 문제들을 해결하기 위해 베이지안 분류기(Bayesian classifier), 신경망(Neural Networks), 유전자 알고리즘(Genetic Algorithms) 같은 다양한 기계 학습 기법을 사용해서 사용자의 프로파일을 동적으로 갱신하는 방법들을 연구했다(Moukas, A., 1996; Lam, W. et al., 1996; Pazzani, M. and D. Billsus, 1997; Tan, A. and C. Teo, 1998). 그러나 이러한 연구들은 과대 특정화(over-specialization)라는 문제점이 있다. 이것은 사용자가 단지 자신의 프로파일에 명시적으로 나타난 정보들에 국한되어 제공 받을 수 있다는 것이다. 게다가 사용자의 기호 정보가 끊임없이 지속적으로 바뀌거나 빈번하게 바뀌는 경우 사용자 기호정보에 대한 프로파일의 복잡성으로 인해 적합하지 못하다. 또한 많은 사용자들은 로그인과 같은 추가적인 행동 없이 사용자 또는 장치를 식별하고 개인화된 검색 결과를 받아보길 원한다. 이것은 로그인을 통한 식별과정 없이 각 사용자(또는 장치)를 식별하고, 사용자의 검색 행동에 대한 지속적인 관찰과 사용자의 묵시적인 피드백을 통해 개인화된 검색 결과를 제공해야 한다는 것을 의미한다.

본 논문에서는 사용자 기호정보의 가변성에 대응하고, 과대 특정화 문제를 해결하기 위해 본 저자들이 이전에 수행했던 적응형 사용자 기호정보 파일기법(Jeon, et al., 2008)을 확장하고, IP address 기반의 장치식별과 묵시적 사용자 피드백을 활용한 개인화 검색을 제공함으로써 사용자가 원하는 결과를 제공하도록 하는 개인화 정보검색시스템인 PIRS(Personalized Information Retrieval System)을 제안한다. 또한 불필요한 검색 결과를 효과적이고, 간단하게 줄일 수 있는 방법 중 하나인 디렉토리 서비스를 응용해 각 검색 결과에 적용했다. 즉, 검색 결과에 대해 실시간 디렉토리 분류 기법을 이용한 필터링을 개인화에 활용함으로써 사용자에게 전달되는 검색 결과의 수를 줄이고 검색 결과에 대한 사용자의 검색 만족도를 개선했다. 이전에 수행했던 작업에서는 각 사용자 기호정보파일의 갱신이 누적된 검색 로그 데이터를 활용한 반면 본 논문에서는 사용자가 웹문서를 클릭하

고 벗어날 때 갱신하도록 함으로써 변화된 기호정보를 보다 빠르게 적용하도록 했으며, 또한 도메인별 유사 사용자 및 전문가의 검색을 실시간으로 수행함으로써 전체 성능이 저하되는 문제를 해결하기 위해 유사 사용자 검색 및 전문가 검색을 오프라인(off-line)으로, 주기적으로 수행 하도록 했다.

본 논문의 구성은 다음과 같다. 먼저 제 2장에서는 기존의 사용자 프로파일 기반 개인화 검색 및 그에 대한 관련 연구에 대해 살펴보고, 제 3장에서는 본 논문에서 제안한 시스템의 구조와 각 구성요소에 대해 자세히 살펴본다. 제 4장에서는 적응형 사용자 프로파일과 검색 결과에 대한 실시간 필터링에 대해 자세하게 설명한다. 제 5장에서는 본 논문에서 제안한 시스템의 성능 평가를 위해 수행한 웹문서의 실시간 필터링 성능 및 사용자 만족에 대한 실험 결과를 기술한다. 마지막 절에서는 제안하는 기법과 시스템에 대해 간략히 요약하고, 향후 작업에 대해 기술한다.

2. 관련 연구

2.1 사용자 프로파일 기반 개인화 정보검색

사용자 프로파일 기법은 개인화 시스템에서 주로 사용되어 온 방법이다(Wang, J. et al., 2006). 일반적으로 개인화 정보검색에서 사용자 프로파일을 사용하는 것은 사용자가 사용한 질의어, 웹문서 방문순서, 웹문서 내에서의 사용자의 행동 또는 방문한 웹문서에 대한 텍스트 분석 자료 등을 저장함으로써 검색 성능을 향상시키기 위한 것이다. 즉, 사용자 프로파일은 질의어를 포함하는 많은 양의 검색 결과를 제공하기 위해서가 아니라 사용자가 원하는 정보를 제공하기 위해 사용된다. 사용자 프로파일의 구성방법은 매우 다양하다. 대부분의 많은 연구들에서는 단순하게 사용자가 사용한 질의어를 벡터로 표현하거나, 사용자가 방문한 웹문서를 분석한 용어(term)와 각 용어에 대한 가중치의 쌍으로 구성한다. 최근에는 사용자 기호정보의 불일치를 해결하기 위해 다양한 방법이 연구되었는데, 기호정보를 장기(long-term)과 단기(short-term)으로 구분해서 구성하는 방법(Sugiyama, K. et al., 2004), 기호정보를 도메인에 따라 구성하는 방법(Chen, T. et al., 2007), 프로파일을 계층적 구조로 구성하는 방법(Nanas, N., V. Uren and A. de Roeck, 2003), 일반적으로 함께 발생하는 용어들을 사용자 프로파일에 그래프로 표현하는 방법(Koutrika, G. and Y. Ioannidis, 2005), 그리고, 온톨로지를 사용해서 사용자의 다양한 상황 및 의미적 요소를 함께 고려하려는 시도들도 있었다(Trajkova, J. and S. Gauch, 2004; Golemati, M. et al., 2007; Robal, T. and A. Kalja, 2007; Stan, J. et al., 2008).

최근의 많은 연구들이 사용자의 묵시적 피드백을 이용해서 사용자 프로파일을 동적으로 갱신하려는 시도를 했다. 과거에 수행한 질의어들과 브라우징 히스토리와 같은 사용자의 다양한 행동들을 분석하고 이를 적용함으로써 사용자 기호정보의 변화에 적응하는 것이다(Nanas, N. et al., 2003; Sugiyama, K. et al., 2004; Trajkova, J. and S. Gauch, 2004; Speretta, M. and S. Gauch, 2005; Zayani, C. et al., 2006; Wang, J. et al., 2006; Stermsek, G. et al., 2007; Chen, T. et al., 2007). 이러한 연구들은 공통적으로 사용자의 기호정보를 자동으로 습득하고 학습하도록 하며, 사용자의 묵시적인 피드백을 이용해서 사용자 프로파일을 동적으로 갱신하도록 한다. 비록 몇몇 실험들을(Kelly, D. and J. Teevan, 2003; Fox, S. 2005) 통해 사용자의 명시적인 피드백에 비해 그 정확도가

떨어진다는 것이 증명되었으나 사용자 간섭 없이 이루어진다는 측면에서의 이점으로 인해 많은 연구자들이 적용하고 있다. 많은 연구자들은 그 정확도를 향상시키기 위해 노력하고 있지만 쉽지 않은 것이 사실이다. 또한 지속적인 기호정보의 변화 또는 기호정보가 빈번하게 발생하는 경우에는 적합하지 않은 단점이 있으며, 사용자의 카테고리별 검색 활동 감시가 용이 하지 않다. 또한 이전에 방문한 사이트를 찾기가 쉽지 않다. 이것은 이전에 방문한 사이트를 재방문하기 위해서 해당 사이트를 찾기 위한 노력이 반복되어야 하는 것을 의미한다. 본 논문에서 제안하는 PIRS에서는 사용자 프로파일을 계층적 구조로 구성함으로써, 사용자의 카테고리별 웹문서 검색 동작을 보다 효과적으로 추적(track) 할 수 있고 사용자의 기호 변화를 적절하게 알아낼 수 있도록 하였다.

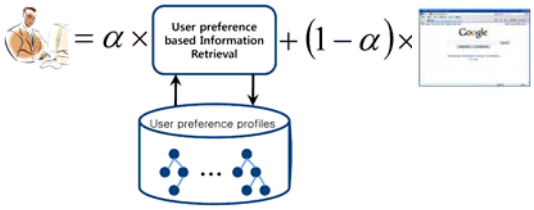
2.2 개인화 검색을 위한 검색 결과의 분류 및 필터링

기존의 개인화 정보검색 시스템들은 각 사용자의 검색 만족도를 향상시키기 위해 다양한 시도를 하고 있지만 쉽지 않은 것이 사실이다. 왜냐하면 검색의도의 불확실성 때문에 사용자의 검색의도를 인지하기 쉽지 않기 때문이다. 검색 의도의 불확실성은 동일한 사용자가 동일한 질의어를 사용하더라도 다른 검색 결과를 원할 수 있다는 것이다. 예를 들면 “java”라는 질의어를 입력했을 때 사용자는 컴퓨터 프로그래밍언어로서의 java를 찾기도 하고 또 때로는 자바산 커피를 찾기 위해 입력하기도 하며 또는 인도네시아의 자바섬을 찾기 위해서 같은 질의어를 사용하기도 한다. 즉, 이러한 검색 의도의 불확실성은 질의어의 중의성 또는 애매모호성에서 비롯된다. 게다가 질의에 사용된 단어의 수가 적을수록 이러한 불확실성은 가중될 것이다. 이러한 불확실성을 줄이고 사용자에게 보다 관련성이 높은 결과를 제공하기 위해 검색 결과에 대한 계층적 분류를 통해 개인화된 온톨로지를 구축하거나(Singh, A. and K. Nakata, 2005; Sieg, A. et al., 2007), 그룹화된 검색 결과를 제공하기도 하고(Kaki, M. 2005), 검색 결과에 대한 사용자의 행동 관찰 및 클릭 정보(clickthrough information)(Joachims, T. 2002)과 같은 특징 데이터를 수집하고 이에 대한 학습을 하기도 한다(Agichtein, E. and Z. Zheng, 2006). 이러한 연구들은 사용자의 묵시적 피드백을 적극 활용함으로써 각 사용자에게 보다 관련된 문서들을 상위에 표시할 수 있도록 했다. 그러나 이들은 사용자의 명시적인 간섭이나 정보를 요구하기도 하며, 불필요한 그룹 결과를 사용자에게 제공하거나, 여전히 관련이 적은 웹 문서들을 사용자에게 전달하기도 하기 때문에 사용자가 원하는 결과만을 제공한다고 보기 어렵다. 본 논문은 이러한 불확실성을 최소화 하고, 관련이 적은 웹 문서를 제공하지 않으며, 사용자의 검색 만족도를 향상시키기 위해 검색 결과에 대해 실시간 분류기법을 이용한 필터링을 수행하였다. 또한 추천 시스템에서 자주 사용되는 협업 필터링 기법(Wang, J. et al., 2006; Naderi, H. and B. Rumpler, 2006, 연철 et al., 2008; Jenu S. et al., 2008)을 이용해 사용자의 기호 정보가 끊임없이 지속적으로 바뀌거나 빈번하게 바뀌는 상황에 적응하였다.

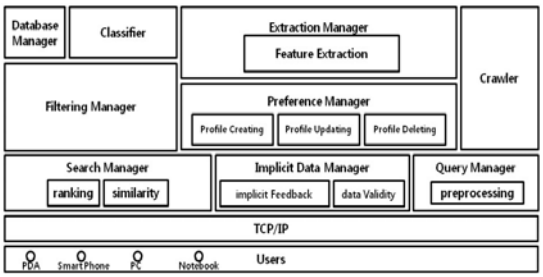
3. 시스템

3.1 시스템 구조

본 논문에서 제안하는 PIRS의 가본개념은 <그림 1>과 같다.



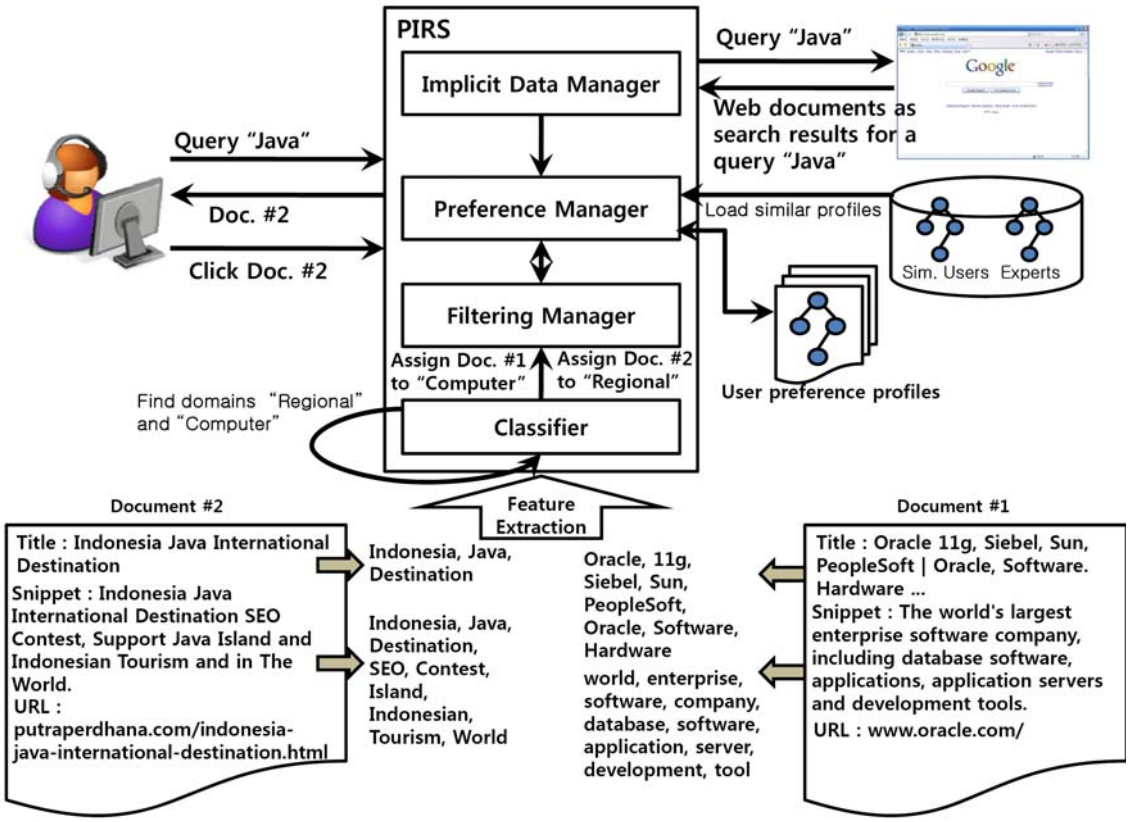
<그림 1> PIRS(Personalized Information Retrieval System)의 기본개념



<그림 2> PIRS의 시스템 구조

<그림 1>에서 보는 것처럼, 제안하는 시스템은 주어진 질의에 대해 사용자 기호정보 프로파일에 기반한 검색을 하고 동시에, 구글과 같은 기존 검색 엔진을 통한 검색 결과를 함께 제공한다. <그림 1>에서 α는 반영요소(reflection factor)로 사용되며, 기존 검색엔진과 사용자 기호정보 프로파일의 검색 결과 수를 결정하는데 사용된다. 초기값은 0.5로 지정된다. 즉, 구글에서 50%, 사용자 기호정보 프로파일에서 50%를 반영한다. 예를 들어, 10개의 검색 결과를 사용자에게 제공한다면 구글에서 5개, 사용자 기호정보프로파일을 통해 5개의 검색 결과가 추출되어 사용자에게 제공된다. 사용자 기호정보 프로파일로부터 추출된 검색 결과에 대한 사용자의 묵시적 피드백이 많아질수록, α의 값은 증가한다. 즉, 사용자 기호정보 프로파일의 반영 비율이 늘어난다.

<그림 2>는 PIRS의 시스템 구조를 나타내며, 9개의 Manager module로 구성되었다. Search Manager는 기호정보를 참조해 검색된 결과에 대한 순위를 재지정하며, 주어진 도메인과 질의에 대해 유사 성향의 다른 사용자들을 찾기 위해 사용자간, 질의간 유사도 계산을 수행한다. Crawler는 사용자의 질의에 대한 구글이나 다른 검색 사이트로부터 웹 문서 검색 결과를 수집한다. Query Manager는 사용자 질의에 대해, 스테밍(stemming), 불용어 제거(stopwords removal), 색인(indexing)과 같은 전처리 과정을 수행한다. Classifier는 사용자의 질의에 대한 검색 결과가 선택된 도메인에 해당하는지를 판단하거나, 도메인 선택 없이 질의 했을 때 적절한 도메인을 찾거나, 또는 도메인 정보가 전혀 없을 때 사용자가 클릭한 결과에 대해 실시간으로 가장 적절한 도메인을 찾는다. Database Manager는 각 장치에 대한 기호정보파일의 reflection factor를 관리한다. Implicit Data Manager는 사용자의 묵시적 피드백을 수집하고, 유효한 피드백 데이터들을 판단하며, 사용자가 만족하는 유효한 데이터에 대해 Preference Manager를 통해 기호정보파일에 적용한다. Extraction Manager는 각 검색 결과에 대한 도메인 분류를 위해 각 검색 결과의 제목과 설명(snippet), url에서 특징을 추출한다. 추출된 특징은 Classifier에 의해 사용되며, 사용자가 도메인 선택 없이 질의한 검색 결과에 대한 클릭 시 적절한 도메인을 찾을 때 또는 사용자가 선택한 도메인에 해당하는 검색 결과들을 사용자에게 제공할 때 사용된다. Filtering Manager는 사용자 기호정보파일로부터 동일 도메인에 대해 수행한 유사 질의 및 선택 결과를 찾고, 가중치 비율에 따라 사용자에게 보여줄 문서의 수와 순위를 결정한다. Preference Manager는 IP address별로 생성되며, 기호정보를 저장하고 있는 기호정보파일에 대한 생성, 수정, 삭제와 같은 관



<그림 3> "Java" 질의어에 대한 PIRS의 처리절차

리 기능을 한다.

3.2 시나리오

<그림 3>은 시스템에 대한 이해를 돕기 위한 동작 시나리오다. 사용자는 인도네시아에 있는 JAVA 지역을 여행하기 위해 정보를 얻으려 한다. 사용자가 도메인 선택 없이 "JAVA"를 질의어로 입력하면, PIRS 시스템은 적절한 도메인 정보를 찾기 위해 사용자 기호정보파일과 각 도메인별 유사 사용자들의 기호정보파일, 각 도메인별 전문가들의 기호정보파일을 활용한다. 즉, 세 종류의 기호정보파일 리스트에서 각 도메인에 대해 "JAVA"와 동일하거나 유사한 질의를 사용한 빈도에 따라 확률값을 계산하고, 확률값에 따라 적절한 도메인을 찾는다. 사용자 기호정보파일로부터 "Regional" 도메인 정보를 찾고, 유사 사용자와 전문가 기호정보파일로부터 "Computer" 도메인 정보를 찾은 경우, 순위를 결정할 때 "Regional"에 속하는 문서들이 상위에 위치하며, "Computer"에 속하는 문서들이 하위에 위치하게 된다. 도메인이 결정되면 PIRS 시스템은 구글로부터 얻어진 검색 결과들 중 상위 50개를 수집한다. 시스템은 각 검색 결과에 대해 title, snippet, url에서 명사들을 추출하

고, 이를 기반으로 각 검색 결과들을 각 도메인에 할당하며, 두 도메인 모두에 포함되지 않는 결과들은 제거된다. PIRS 시스템은 사용자 기호정보파일에서 이전에 수행했던 검색 중 유사한 질의를 찾고 사용자가 선택했던 결과들을 추출한다. 또한 각 도메인에 대해 유사 성향의 사용자가 선택한 문서들과 각 도메인의 전문가들이 선택한 문서들을 함께 결과로 사용자에게 제공한다. 사용자 자신의 기호정보파일 및 여러 정보 소스로부터 추출된 문서들을 재순위화하고 이를 사용자에게 제공한다. 검색 결과들에 대한 사용자의 클릭은 묵시적 피드백으로 사용되며, 사용자가 선택한 검색 결과 중 관련된 문서들에 대해 반영 비율값과 사용자 기호정보파일의 가중치값들이 재계산되어 반영된다.

4. 개인화 정보검색 기법

4.1 적응형 사용자 프로파일

각 사용자 기호정보 프로파일은 IP address 별로 생성되고, <그림 4>와 같은 계층적 구조로 이루어 졌으며, <그림 5>와 같은 XML 파일로 저장된다. <그림 4>에서 사용된 인자들은 다음과 같다.

U : 도메인의 집합

U = {Domain1, Domain 2, …, Domain n}

$Domain_n$: 사용자가 선택한 문서들이 속한 도메인

$Domain_n = <T_1 : WD_nT_1, T_2 : WD_nT_2, \cdots, T_m : WD_nT_m>$.

T : 사용자 질의에 대한 term vector

$T = <t_1, t_2, t_3, \cdots, t_n>$

t_i = 사용자가 사용한 질의의 i번째 term

D : 사용자가 선택한 문서들의 집합

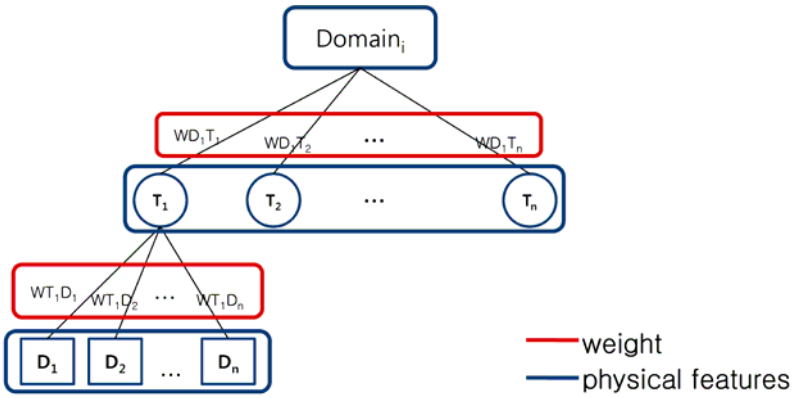
$D = \{D_1, D_2, D_3, \cdots, D_n\}$

D_i : 사용자가 선택한 i번째 문서

WD_iT_j : 도메인 D_i 에 대해 질의 T_j를 사용할 확률

WT_iD_j : 질의 T_i에 대해 문서 D_j를 선택할 확률

이때, 상위 구조는 사용자가 선택한 도메인(또는 디렉토리)이며, 하위구조는 사용자가 사용한 질의 및 선택한 검색 결과의 쌍으로 구성된다. 사용자 프로파일을 계층적 구조로 유지함으로써 동적으로 변화하는 사용자의 기호정보에 보다 효율적으로 대응할 수 있다. 또한 빈번하게 변화하는 사용자의 기호정보 변화에 대응하기 위해 협업 필터링(collaborative filtering) 기법을 활용해 사용



<그림 4> 사용자 프로파일 구조

```
<profile>
<domain_info>
<domain count="23" name="Sports">
<term count="8" name="jisung" wDt="0.348">
<document count="3" wtd="0.375">
<documentURL>http://liverex.tistory.com/481</documentURL>
</document>
<document count="1" wtd="0.125">
<documentURL>http://www.youtube.com/watch?v=Ph2XnbHUgm0</documentURL>
</document>
<document count="1" wtd="0.125">
<documentURL>http://rkiller.tistory.com/15</documentURL>
</document>
<document count="1" wtd="0.125">
<documentURL>http://joongangdaily.joins.com/article/view.asp?aid=2916036</documentURL>
</document>
<document count="2" wtd="0.25">
<documentURL>http://rachael.tistory.com/70</documentURL>
</document>
</term>
<term count="15" name="worldcup" wDt="0.652">
<document count="3" wtd="0.2">
<documentURL>http://www.fifa.com/matches/index.html</documentURL>
</document>
<document count="6" wtd="0.4">
<documentURL>http://worldcup.sbs.co.kr</documentURL>
</document>
<document count="3" wtd="0.2">
<documentURL>http://www.worldcupblog.org</documentURL>
</document>
<document count="3" wtd="0.2">
<documentURL>http://soccernet.espn.go.com/world-cup/?cc=4716&ver=global</documentURL>
</document>
</term>
</domain>
</domain_info>
</profile>
```

<그림 5> 사용자 프로파일 예

자가 선택한 도메인 내에서 가장 유사한 성향을 지닌 사용자와 가장 활발한 활동을 한 전문가를 검색하고 사용자 질의와 유사한 이들의 질의 및 검색 결과들을 추출한 뒤 이를 사용자에게 제공함으로써 사용자가 처음 입력하는 질의이거나 급작스런 기호정보의 변화에 대처할 수 있다. 또한 사용자 질의에 대한 기존 검색엔진의 결과를 사용자에게 함께 제시함으로써 콜드 스타트(cold start) 문제를 해결 하도록 했다.

사용자 프로파일의 갱신은 동적, 자동적으로 이루어지며, 사용자의 묵시적 피드백을 이용해 긍정과 부정을 판단한다. 이를 위해(S. Fox, 2005)의 결과를 이용했으며, 다른 여러 연구들(D. Kelly, and J. Teevan, 2003; S. Fox, 2005)에서 사용된 많은 특징들 중에서 일부를 사용했다.

사용자가 클릭한 검색 결과의 긍정여부를 판단하기 위해 “문서에 머문 시간”, “문서의 길이”와 “탈출조건”과 같은 묵시적 피드백 특성들을 사용해서 자동으로 판단하고, 사용자가 검색리스트로 돌아오거나 윈도우를 닫는 행동처럼 해당문서를 벗어날 때, 기호정보파일을 갱신하였다. 논문에서 사용한 규칙은 다음과 같으며, 사용자가 선택한 문서가 다음의 조건을 만족하면 긍정으로 인식한다.

$$\left(\begin{array}{l} DifferenceSec > 58.4 \ \wedge \\ DurationSec > 27.1 \ \wedge \\ DocumentLeng > 225 \ \wedge \\ ExitType\ is\ Not\ Back\ to\ Result\ List \ \wedge \\ AbsolutePosition < 3.45 \\ ImageCount > 1 \end{array}\right) \vee (AddFavorite \vee Print \vee FileSave)$$

<표 1> 묵시적 피드백 특성

조건	설명
문서에 머문 시간 (DurationSec)	문서 다운로드가 완료 후부터 문서를 벗어날 때까지의 시간
검색리스트로 돌아오는데 걸린 시간 (DifferenceSec)	검색 결과의 한 문서를 접근한 후, 다시 검색 리스트로 돌아오는데 까지 걸린 시간
문서의 길이 (DocumentLeng)	선택한 문서의 문자 수
이미지의 수 (ImageCount)	선택한 문서에 포함된 이미지의 수
탈출조건 (ExitType)	문서를 벗어나는 조건
문서위치 (AbsolutePosition)	검색 결과 내에서의 위치
북마크 추가여부 (AddFavorite)	해당 문서를 즐겨찾기에 추가했는지 여부
출력여부 (Print)	해당 문서의 출력 여부
파일 저장 여부 (FileSave)	해당 문서에 포함된 파일의 저장 여부

<표 1> 은 본 논문에서 사용된 묵시적 피드백 특성들을 나타낸 것이다. 본 논문에서는 이러한 특성들을 모든 사용자에게 일괄적으로 적용했다. 또한 각 사용자의 기호정보파일 갱신은 해당문서를 벗어날 때 이루어지며, 해당 도메인, 사용한 질의어, 선택된 문서와 같은 정보들을 이용해서 사용자 기호정보파일의 각 가중치값을 갱신한다. 도메인과 사용된 질의간의 가중치인 WDT와, 사용된 질의와 선택된 문서간 가중치인 WTD는 다음의 식 (1)과 식 (2)를 이용해 계산된다.

$$WDT = \frac{Query\ frequency}{Domain\ frequency} \quad (1)$$

$$WTD = \frac{Document\ frequency}{Query\ frequency} \quad (2)$$

이때, Domain frequency는 사용자가 선택한 문서가 속한 도메인의 빈도수, 즉, 사용자가 해당 도메인을 선택한 빈도수를 의미하며, Query frequency는 사용자가 해당 도메인에서 해당 질의를 사용한 빈도수를 의미한다. Document frequency는 해당 도메인에서 해당 질의를 사용했을 때, 사용자가 해당 문서를 선택한 빈도수를 의미한다.

기호정보파일의 가중치값들이 재계산되면, 기호정보를 구성하는 각 질의 와 선택 문서들의 삭제 여부를 결정한다. 즉, WDT가 임계값(0.05) 이하 이면 해당 질의를 삭제하며 이때, 해당 질의에 대한 선택된 문서들도 함께 삭제된다. 또한 WTD가 임계값(0.1)이하일 경우 해당 문서를 삭제한다. 즉, 기호정보파일은 갱신 때마다 새로운 문서나 질의가 추가 될 수도 있으며, 기존 질의나 문서가 삭제되기도 한다.

4.2 검색 결과의 실시간 필터링

본 논문에서는 보다 향상된 개인화 검색을 위해 웹 검색 결과에 대한 실시간 필터링을 수행하였다. 즉 사용자가 질의시 선택한 도메인에 해당하는 검색 결과들만 사용자에게 제공함으로써 사용자에게 전달되는 관련이 적은 문서의 수를 줄였다. 다시 말하면, 검색 시 도메인을 선택하는 행동은 사용자의 검색 의도를 명시적으로 표현하는 것이며, 해당 도메인과 관련된 검색 결과들만 제공하는 것은 이러한 사용자의 요구에 응대하는 것이다.

검색 결과들에 대한 실시간 필터링을 위해 구글에서 제공하는 디렉토리 서비스를 활용했다. 먼저, 구글에서 제공하는 디렉토리 서비스의 각 도메인(또는 디렉토리)에 속하는 일부 사이트 정보(제목, 설명(snippet), url)를 수집했고, 색인과정을 통해 용어들을 추출했다. 또한 색인 작업 직후, 각 용어

```
RealTimeFiltering (query, domain)
1     GoogleDocs ← receive retrieved web documents from Google
2     NoiseTerms ← load noise terms of the domain
3     UniqueTerms ← load unique terms of the domain

4     for each document d in GoogleDocs do
5        d = removeNoiseTerms(d);
6
7        if d.Title or d.Snippet include unique term t in UniqueTerms then
8           continue;
9        else
10          domains[] = CosSim(d);   // calculate cosine similarity value for each domain
11          dom = findMAXDomain(domains);   // find domain dom with max sim value in domains
12          if (domain = dom) and (difference of sim values between first domain and second
      domain ≤ threshold value) then
13           continue;
14          else
15             remove d from GoogleDocs
16
             end if
          end if
17    end for
```

<그림 6> 실시간 필터링을 위한 의사코드

에 대한 TF(term frequency)와 IDF(inverse domain frequency in system)를 계산했다. 즉, 각 도메인을 문서집합 내의 각 문서로 가정하고 각 검색 결과 문서를 질의라고 여겨 그 유사도를 계산했다.

$$Sim(Q,\ D_i) = \frac{\sum_i w_{Q,j} \cdot w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2}\sqrt{\sum_i w_{i,j}^2}} \quad (3)$$

$$w = \frac{tf}{\sum_k tf_k} \times \left(\log\left(\frac{N}{df}\right)+1\right) \quad (4)$$

이때, 식 (3)에서 w_{Qj}는 질의 Q에 나타난 j번째 용어에 대한 가중치값을 의미하며, w_{ij}는 i번째 문서에 나타난 j번째 용어에 대한 가중치 값을 의미한다. 각 용어에 대한 가중치값은 식 (4)를 통해 계산되며, 식 (4)의 tf는 용어가 나타나는 빈도수를 의미하며, df는 용어가 나타나는 도메인의 수를 나타내는 도메인 빈도수를 의미하고, N은 전체 도메인의 수를 나타낸다.

정보 검색에서 자주 사용되는 식 (3), 식(4)의 cosine TF · IDF 유사도 기법을 활용해 각 검색 결과가 주어진 도메인에 해당하는지를 실시간으로 판단하였다. 이를 위해 각 도메인에 대해 수집된 사이트 정보를 이용해서 트레이닝과 테스트를 수행했고, unique term과 noise term을 정의했다. Unique term이란 특정 도메인에서만 나타나는 용어를 의미한다. Noise term은 다른 도메인에 대한 가중치값(TF · IDF)이 해당 도메인에 대한 가중치값 보다 크거나 가장 큰 가중치값을 가지는 도메인과 두 번째 큰 값을 지니는 도메인간 차이가 일정 임계값 이하인 용어를 의미한다. 즉, 용어의 가중치값이 해당 도메인에 대해 가장 큰 값이어야 하며, 다른 도메인에 비해 압도적으로 커야 한다.

<그림 6>은 검색 결과에 대한 실시간 필터링을

위한 의사코드를 나타낸 것이다. 검색 결과에 대한 실시간 필터링은 사용자 질의와 도메인 정보를 받는 것부터 시작하며, 주어진 질의에 대한 구글의 검색 결과를 받는다. 주어진 도메인에 대한 noise term과 unique term을 로드한 후, 구글로부터 전달된 각 검색 결과 d에서 noise term을 제거한다. 이후, 각 검색 결과 d에 대해 unique term의 포함 여부를 확인하고, 만약 포함한다면 해당검색 결과는 최종 결과로 남게 되며, 그렇지 않으면 각 도메인에 대한 검색 결과 d의 코사인유사도 값들을 계산한다. 계산된 유사도값들 중 가장 큰 값을 지니는 도메인이 주어진 도메인과 같고, 두 번째 큰 값과의 차이가 일정 임계값보다 크면 최종 결과에 남게 된다. 그러나 두 조건 중 하나라도 만족하지 않으면 최종 결과에서 제거된다.

도메인 정보가 제공되지 않는 경우, 실시간 필터링을 수행하기 위해, 시스템은 주어진 질의에 적절한 도메인을 찾는 선행 작업을 수행한다. 시스템은 적절한 도메인 정보를 찾기 위해 사용자 기호정보파일과 각 도메인별 유사 사용자들의 기호정보파일, 각 도메인별 전문가들의 기호정보파일을 활용한다. 즉, 세 종류의 기호정보파일 리스트에서 각 도메인에 대해 주어진 질의와 동일하거나 유사한 질의어를 사용한 빈도에 따라 확률값을 계산하고, 확률값에 따라 적절한 도메인을 찾으며, 동일한 확률값을 갖는 경우 모두 선택한다. 이때 사용자 자신의 기호정보파일로부터 찾은 도메인 정보와 유사 사용자와 전문가의 기호정보파일로부터 찾은 도메인 정보는 병합되지 않는다. 이것은 유사 사용자와 전문가의 기호정보파일을 참고하는 이유와 동일하며 두 가지 사항을 고려하기 위한 것이다. 사용자 기호의 급작스러운 변화에 대응하고, 과대 특정화를 해결하기 위해 도메인별 유사 사용자들의 기호정보파일과 도메인별 전문가들의 기호정보파일을 참조함으로 다양한 도메인에 대해 접근 할 수 있도록 했다. 이때, 발생 가능한 세 가지 상황이 있다. 첫째, 동일한 질의 또는 유사한 질의를 찾지 못해 적절한 도메인을 찾지 못한 경우, 둘째, 단 하나의 도메인을 찾는 경우, 마지막으로, 둘 이상의 도메인을 찾는 경우가 그것이다. 첫 번째 경우는 실시간 필터링 수행 없이 구글의 결과들을 사용자에게 전달하며, 두 번째 경우는 선택된 도메인을 사용하며, 마지막 경우는 선택된 모든 도메인을 사용한다.

적절한 도메인을 찾기 위해 각 도메인에 대한 유사 질의집합 SQi을 찾는다. 유사 질의 질의집합은 다음과 같이 정의된다.

$$SQ_i = \{ \forall_T | distance(Q,T) \le 1 \}$$

이때, SQ_i는 i번째 도메인에 대한 유사 질의집합, distance는 사용자 질의와 기호정보파일의 i번째 도메인에 대해 사용한 각 질의간 거리를 의미하며, 이러한 거리가 1이하인 경우 유사 질의로 판단한다. 사용자 질의와 기호정보파일의 각 질의간 거리는 다음의 식 (5)와 같은 유클리디안 거리(Euclidean Distance)를 이용해서 측정한다.

$$distance(Q,T) = \sqrt{\sum_{i=1}^{n}(qt_i - t_i)^2} \qquad (5)$$

각 도메인에 대한 확률값은 다음의 식 (6)과 같이 계산한다.

$$\Pr(D_i) = \frac{TFr_i}{TFr} = \frac{\sum_j tfr_{ij}}{\sum_p \sum_q tfr_{pq}} \qquad (6)$$

이때, TFr은 전체 유사 질의 빈도수의 합, TFr_i은 i번째 도메인 Di의 유사 질의 빈도수의 합, tfr_{ij}는 i번째 도메인에 속한 j번째 유사 질의 빈도수, tfr_{pq}는 p번째 도메인에 속한 q번째 유사 질의 빈도수를 의미한다.

다음의 식 (7)을 통해 적절한 도메인을 결정한다.

$$Domain = max Pr(D_i) \quad (7)$$

4.3 사용자 검색과 협업 필터링

유사 사용자와 전문가는 협업 필터링을 사용함으로써 과대 특정화를 해결하기 위해 활용되며, 유사 사용자와 전문가 리스트는 Filtering Manager에 의해 도메인별로 각각 최대 5명씩 주기적으로(1일) 사전에 검색되어 DB Manager를 통해 데이터베이스에 저장된다.

4.3.1 유사 사용자

유사 사용자 검색은 도메인 단위로 이루어지며, 각 질의어 집합을 벡터공간 모델로 표현하고, 해당 도메인에서 두 사용자가 사용한 질의 집합에 대해 식 (8)과 같은 유클리디안 거리로 계산 한다.

$$Dis(T, T') = \sqrt{\sum_{i=1}^{n}(t_i - t_i')^2} \quad (8)$$

4.3.2 전문가

전문가는 사용자와의 유사 성향 여부와 상관없이 사용자가 선택한 도메인 및 사용한 질의에 대해 많은 양의 문서를 작성하거나 선택한 사용자들을 의미한다. 본 논문에서는 전문가 선택의 기준으로 해당 도메인에 대해 작성하거나 선택한 문서의 양이 많은 사용자로 정했으며 상위 5명을 추출했다.

4.3.3 협업 필터링

협업 필터링은 과대 특정화를 해결하기 위해 해당 도메인에 대한 유사 성향의 사용자들의 기호정보파일과 해당 도메인의 전문가들의 기호정보파일을 사용하며, Filtering Manager에 의해 수행된다. 협업 필터링은 도메인별 유사 성향의 사용자와 전문가를 찾고, 각 사용자 기호정보파일에서 해당 도메인 및 유사 질의를 찾아 연결된 문서들을 사용자에게 제공한다. 유사 질의는 식 (5)를 사용해서 가장 작은 값을 지니는 질의를 찾으며, 연결된 문서들을 추출한다. 이때, 문서들의 순위는 질의와 문서간 가중치값, 즉, WTD에 따른다. 또한 추출되는 문서 수는 반영비율값에 의해 전체 기호정보파일로부터 추출되는 문서의 수 중에서 각각 20%를 넘지 않도록 했다. 유사 질의를 찾는 과정에서 동일한 값을 지니는 둘 이상의 질의가 있는 경우 해당 문서들을 모두 병합한다. 사용자 기호정보파일에서 추출된 결과들은 기호정보파일에 나타난 질의와 선택된 문서간 가중치에 따라 순위가 조절되며, 구글로부터 추출된 검색 결과들과 사용자 기호정보파일로부터 추출된 검색 결과들은 가중치값과 반영 비율값에 따라 그 수와 순위가 결정된다. 검색 결과들에 대한 사용자의 클릭은 묵시적 피드백으로 사용되며, 사용자가 선택한 검색 결과는 그 진위여부에 따라 반영 비율값과 사용자 기호정보파일의 가중치값들이 재계산되어 반영된다.

5. 실 험

본 절에서는 제안하는 검색 결과에 대한 실시간 필터링과 개인화 검색의 성능 평가를 위해 수행한 몇 가지 실험에 대해 기술하고, 제안하는 PIRS의 성능에 대해 설명한다.

5.1 검색 결과의 실시간 필터링

검색 결과에 대한 실시간 필터링의 성능평가를 위해 구글 디렉토리 서비스의 각 디렉토리별로 사이트 정보들을 수집하고 색인 작업을 수행하였으며, 이들에 대해 트레이닝과 테스트를 수행하였다. 수집된 사이트의 수, unique term 수, noise term 수는 <표 2>와 같다. 또한 색인과정 직후 unique term과 noise term을 정의하였고, 이들은 <그림 7>에 나타난 것처럼 실시간 필터링의 성능향상에 매우 크게 기여했다.

<표 2> 도메인별 사이트 수, unique term 수, noiseTerm 수

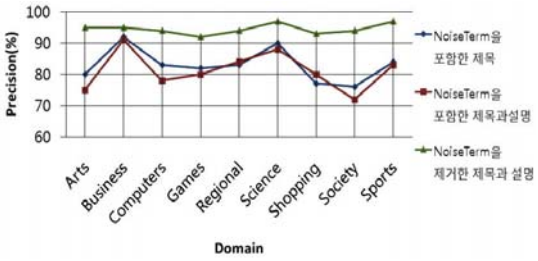
도메인	사이트 수	unique term 수	noise term 수
Arts	29484	13992	50862
Business	52582	26699	32142
Computers	14076	8462	20693
Games	11167	4331	13359
Regional	22666	9130	34931
Science	19764	10062	14440
Shopping	27931	12774	35111
Society	19672	7918	38054
Sports	18217	6534	23929

최적의 성능을 나타내는 방법을 찾기 위해 다음의 세 가지 가능한 경우에 대해 실험을 했으며, 최적의 성능을 나타낸 경우를 검색 결과에 대한 실시간 필터링에 적용했다. 세 가지 경우는 1) 각 사이트에 대해 noise term을 포함한 제목만 사용한 경우, 2) noise term을 포함한 제목과 설명을 사용한 경우, 3) noise term을 제거한 제목과 설명을 사용한 경우다.

식 (9)의 정확도를 사용해서 상위 10개의 검색 결과에 대해 측정했으며, <그림 7>을 통해 알 수 있듯이 noise term을 제거했을 때의 정확도가 그렇지 않았을 때 보다 매우 높다.

또한, 사이트의 제목과 설명을 함께 사용하는 경우와 사이트의 제목만 사용하는 경우는 매우 유사한 성능을 나타냈지만 전체 평균값은 각각 83과 81.2로 사이트의 제목만 사용하는 경우가 약간 좋은 성능을 나타냈다. 이것은 사이트 설명에 사용된 각 용어들이 일반적인 용어인 반면에 제목에 사용되는 용어들은 주로 해당 사이트를 직관적으로 나타낼 수 있는 용어들이기 때문이다. 그러나 사용되는 용어의 수가 매우 적을 뿐만 아니라, 각 용어의 빈도수가 매우 적기 때문에 noise를 제거하는 방법은 적절하지 않다.

$$Precision = \frac{Number\ of\ Retrieved\ Relevant\ Documents}{Number\ of\ Retrieved\ Documents(10)} \quad (9)$$



<그림 7> 각 도메인별 정확도

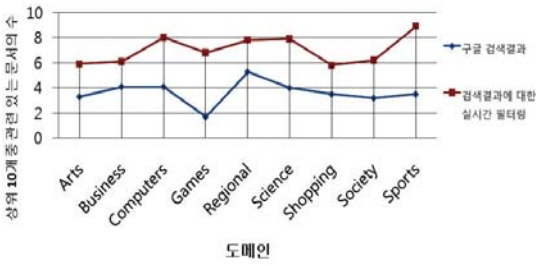
실시간으로 각 검색 결과들이 주어진 도메인에 해당하는지를 판단하기 위해 정보검색 분야에서 가장 널리 사용되는 Cosine TF-IDF 기법을 사용했다. 이때 각 도메인이 하나의 문서로 간주되고, TF 계산시 각 도메인마다 사이트의 수가 다르기 때문에 식 (10)과 같이 각 용어에 대해 정규화 과정(term normalization)을 수행했다.

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{k,j}} \quad (10)$$

이때, n_{ij}는 term t_i가 도메인 d_j에서 나타나는 빈도수 이며, $n_{k,j}$의 합은 도메인 d_j내에 있는 모든 단어의 빈도수 합이다.

Noise term을 제거한 cosine TF · IDF 기법을 적용한 실시간 필터링 기법과 상위 10개의 검색 결과에 대해 식 (4)의 정확도를 이용해서 성능을 평가했다. 질의어의 애매모호성에 대한 성능을 측정하기 위해 하나의 단어로 구성된 질의어들을 실험에 사용했다. 각 도메인마다 java, manchester, sun, jakarta와 같은 10개의 애매모호성을 지닌 질의어를 사용해서 식 (4)의 정확도에 따라 그 정확성을 측정했으며, 동일한 질의어에 대해 일반적인 구글의 검색 결과와 비교했다. 논문에서 제안한 실시간 필터링 기법이 도메인 정보 없이 질의에 적절한 도메인을 찾기 위한 방법이 아니기 때문에, 실험은 도메인 정보가 주어진 상태에서 이루어 졌다. <그림 8>은 각 도메인별로 주어진 10개의 질의에 대한 일반적인 구글 검색 결과와 실시간 필터링 기법간의 정확도를 비교하고 있다.

<그림 8>에 나타난 것처럼 구글 검색 결과에 대한 전체 평균값은 약 4인 반면 제안한 실시간 필터링 기법의 전체 평균값은 약 7로 약 2배에 이른다. 이러한 정확도는 트레이닝과 테스트에 사용되는 문서의 수가 증가함에 따라 보다 향상될 것으로 예상된다. 이것은 구글에서 주어진 질의에 대해 대중적인 결과들이 우선적으로 나타나는 반면 실시간 필터링은 주어진 도메인 정보를 사용함으로써 주어진 도메인과 관련된 정보들이 상위에 표시되도록 하기 때문이다. 예를 들면, “java”라는 질의에 대해 구글은 Computer Language Java가 상위 10개의 검색 결과 대부분을 차지한 반면 실시간 필터링은 주어진 도메인에 따라 “Regional” 또는 “Computer”와 관련된 결과만을 제공하도록 했다.



<그림 8> 실시간 필터링과 구글의 검색 결과 비교

본 논문에서 구글 디렉토리 서비스와 직접 비교하지 않은 것은 실시간 필터링은 사이트 뿐만 아니라 모든 웹 문서를 포함하는 반면에 구글 디렉토리 서비스는 그 대상이 사이트에 국한되기 때문이다.

5.2 개인화 정보 검색

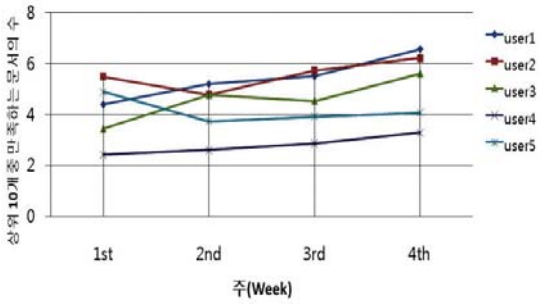
적응형 사용자 프로파일링 기법의 성능을 평가하기 위해, 식 (11)과 같은 정확도를 정의했고, 5명의 사용자에 대해 약 4주간 실험을 수행하였다.

$$SAT = \frac{Number\ of\ Retrieved\ Satisfied\ Documents}{Number\ of\ Retrieved\ Documents(10)} \quad (11)$$

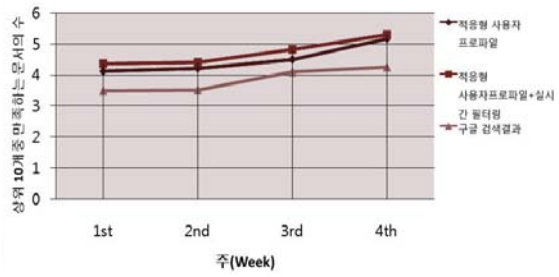
각 사용자는 각 도메인마다 최소 1개의 질의를 수행했으며, 각 검색 결과에 대해 사용자로부터 명시적으로 만족 여부를 묻는 대신(S. Fox, 2005)에서 사용된 결정 트리(decision tree) 조건을 응용해 본 논문에서 정의한 조건식을 사용했다.

논문에서 각 검색 결과에 대한 사용자의 묵시적 피드백을 통해 만족 여부를 판단하였다. <그림 9>는 각 사용자가 약 4주 간 수행한 검색에 대해 상위 10개의 검색 결과에 대한 만족도를 표시하고 있다. <그림 9>에서 나타난 것처럼 대부분의 사용자들은 시간이 경과함에 따라 상위 10개의 검색

결과에 대해 만족하는 문서의 수가 증가한다. 이것은 실험에 참여한 대부분의 사용자들이 이전에 접근했던 문서를 다시 접근하는 비율이 높아진다는 것을 의미하며, 반면에 만족하는 문서의 수가 감소하는 경우는 이전에 접근했던 문서에 대한 재접근 비율이 감소한다는 의미다. 이러한 경우는 검색을 할 때마다 새로운 질의어를 사용하거나 새로운 문서를 찾는 경우 또는 이전의 문서를 재접근하려 하지만 동일한 질의어를 사용하지 않은 경우 등이다.



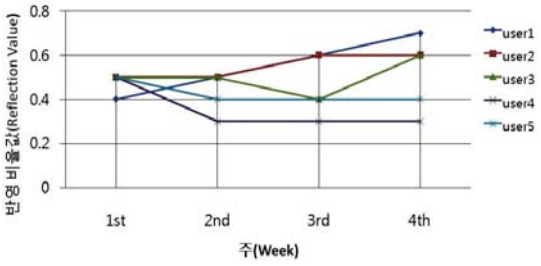
<그림 9> 상위 10개 결과 중 만족하는 문서의 수



<그림 10> 상위 10개의 결과 중 만족하는 문서의 수 비교

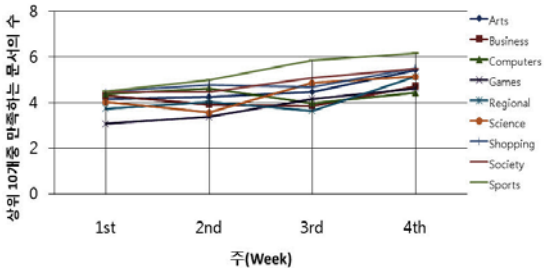
<그림 10>은 실험에 참가한 각 사용자에게 동일한 질의어를 사용해서 적응형 사용자프로파일 기법(Adaptive User Profile)과 검색 결과에 대한 실시간 필터링, 적응형 사용자 프로파일, 그리고 Google에 대해 상위 10개의 검색 결과 중 만족하는 문서의 평균 수를 비교한 것이다. 그림에서 나타나는 것처럼, 검색 결과에 대한 실시간 필터링을 함께 사용함으로써 적응형 사용자 프로파일 기법만 사용했을 때보다 약 5%, 구글 검색 결과에 비해 약 23.2% 향상됐다. 또한 적응형 사용자 프로파일 기법은 구글 검색 결과에 비해 약 17.3 %의 성능이 향상되었다.

<그림 11>은 각 사용자의 반영비율값(reflection value)를 나타내고 있다. 이 값이 낮은 사용자일수록 검색을 수행할 때 마다 다른 질의를 하거나 이전에 접근했던 웹문서에 접근하기 위해 사용한 질의어가 다르다. 반면에 이 값이 높은 사용자들은 이전에 접근했던 문서를 재접근하는 비율이 높으며, 동일한 질의어 사용 빈도가 높다.



<그림 11> 반영 비율값(reflection value)

<그림 12>는 시간 경과에 따른 도메인별 상위 10개의 검색 결과문서 중 평균 관련 문서의 수를 나타낸다.



<그림 12> 도메인별 평균 관련 문서의 수

<표 3>은 구글로부터 검색 결과를 가져오는 수에 따른 각 과정별 소요시간을 나타낸다. 표에서 나타난 것처럼 구글로부터 가져오는 결과의 수와 실시간 필터링 시간은 비례해서 늘어났다. 특히 100개의 검색 결과를 가져오는 경우 50개를 가져올 때 보다 약 14배의 시간이 소요된다. 그러나 다른 과정들은 구글 검색 결과 수에 영향을 받지 않고 수행됐다.

<표 3> 구글 검색 결과 수 변화에 따른 검색 과정별 평균 소요시간

	10	30	50	100
구글검색 추출	282.2	426.9	493.4	816.6
유사도메인 추출	53.5	45.1	50.2	47.8
실시간 필터링	567.3	1390	2219	32727
재순위화	11.9	12.6	11.8	13.3

6. 결론 및 향후 과제

본 논문에서는 사용자에게 보다 개인화된 검색 결과를 제공하기 위해 각 장치마다 계층적 구조로 구성된 사용자 프로파일을 작성했으며, 성능 향상을 위해 검색 결과에 대한 실시간 필터링 기법을 적용하였다. 또한 사용자 프로파일을 갱신하기 위해 묵시적 사용자 피드백을 사용함으로써 사용자의 검색 결과에 대한 명시적 요구를 대체하도록 했다. 이러한 시도들은 실험을 통해 Google 검색에 비해 약 23.2%의 성능 향상이 있음을 확인했다. 이는 만족할 만한 결과이며, 또한 매우 가치 있는 것임을 본 논문은 주장한다.

그러나 좀 더 높은 수준의 만족도나 정확도를 충족시키기 위해서 보다 향상된 실시간 필터링 기법과 함께 묵시적 사용자 피드백 특징 값의 개인화가 적용되어야 할 것이다. 또한 사용자 개인을 식별할 수 있는 새로운 방법이 필요하며, 사용자 뿐만 아니라 사용자가 사용하는 각 장치들에 대한 고려도 필요하다.

참고문헌

연철, 지애띠, 김홍남, 조근식, “효과적인 추천 시스템을 위한 협업적 태그 기반의 여과 기법”, 한국지능정보시스템학회, 한국지능정보시스템학회논문지, 14권 2호(2008), 157~177.

Agichtein, E. and Z. Zheng, “Identifying “best bet” web search results by mining past user behavior”, *In Proc. 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*(KDD-06), (2006), 902~908.

Chen, T., W. Han, H. Wang, Y. Zhou, B. Xu, and B. Zang, “Content recommendation system based on private dynamic user profile”, *In Proc. Intl. Conf. on Machine Learning and Cybernetics*, (2007), 2112~2118.

Fox, S., K. Karnawat, M. Mydland, S. Dumais and T. White, “Evaluating implicit measures to improve web search”, *ACM Transactions on Information Systems*, Vol.23, No.2(2005), 147~168.

Golemati, M., A. Katifori, C. Vassilakis, G. Lepouras, and C. Halatsis, “Creating an ontology for the user profile : Method and applications”, In Proc. 1st RCIS Conf., (2007), 23~26.

Jenu Shrestha, Mohammed Nazim Uddin, and GeunSik Jo, “Combining Collaborative, Diversity and Content Based Filtering for Recommendation System”, 한국지능정보시스템학회, 한국지능정보시스템학회논문지, Vol.14, No.1(2008), 101~115.

Jeon, H. C., T. H. Kim, and J. M. Choi, "Adaptive user profiling for personalized information retrieval", *In Proc. 3rd Intl. Conf. on Convergence and Hybrid Information Technology*(ICCIT 2008), Vol.2 (2008), 836~841.

Joachims, T., "Optimizing search engines using clickthrough data", In Proc. 8th ACM SIGKDD Intl. Conf. on Knolwedge Discovery and Data Mining, (2002), 133~142.

Kaki, M., "Enhancing Web search result access with automatic categorization", Ph.D. dissertation, Department of Computer Sciences, University of Tampere, Tampere, Finland, 2005.

Kelly, D. and J. Teevan, "Implicit feedback for inferring user preference : A bibliography", *ACM SIGIR Forum*, Vol.37, No.2(2003), 18~28.

Koutrika, G. and Y. Ioannidis, "A unified user profile framework for query disambiguation and personalization", *In Proc. Workshop on New Technologies for Personalized Information Access* (PIA2005), (2005) 44~53.

Lam, W., S. Mukhopadhyay, J. Mostafa, and M. Palakal, "Detection of shifts in user interests for personalized information filtering", *In Proc. 19th Intl. ACMSIGIR Conf. on Research and Development in Information Retrieval*, (1996), 317~325.

Moukas, A., "Amalthea : Information discovery and filtering using a multiagent evolving ecosystem", In Proc. 1st Intl. Conf. on The Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM), 1996.

Naderi, H. and B. Rumpler, "PERCIRS : a system to combine personalized and collaborative information retrieval", *Journal of Documentation*, Vol.66, No.4(2006), 532~562.

Nanas, N., V. Uren and A. de Roeck, "Building and applying a concept hierarchy representation of a user profile", *In Proc. 26th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, (2003), 194~204.

Pazzani, M. and D. Billsus, "Learning and revising user profiles : The identification of interesting web sites", *Machine Learning*, Vol.27(1997), 313~331.

Robal, T. and A. Kalja, "Applying user profile ontology for mining web site adaptation recommendations", *In Proc. ADBIS 2007. LNCS*, Vol.4690(2007), 126~135.

Sieg, A., B. Mobasher, and R. Burke, "Representing context in web search with ontological user profiles", In Proc. 6th Intl. and Interdisciplinary Conf. on Modeling and Using Context, LNCS Vol.4635(2007), 439~452.

Singh, A. and K. Nakata, "Hierarchical classification of web search results using personalized ontologies", *In Proc. 3rd Intl. Conf. on Universal Access in Human Computer Interaction*, (2005).

Speretta, M. and S. Gauch, "Personalized search based on user search histories", *In Proc. IEEE/ACM Intl. Conf. on Web Intelligence*, WI '05, (2005), 622~628.

Stan, J., E. Egyed Zsigmond, A. Joly, and P. Maret, "A user profile ontology for situation-aware social networking", In Proc. 3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence, 2008.

Stermsek, G., M. Strembeck, and G. Neumann, "User profile refinement using explicit user interest modeling", In Proc. 37. Jahrestagung der Gesellschaft f¨ur Informatik (GI). Lecture Notes in Informatics (LNI), Vol.

109, 289~293, (2007).

Sugiyama, K., K. Hatano, and M. Yoshikawa, "Adaptive web search based on user profile constructed without any effort from users", In Proc. 13th Intl. Conf. on World Wide Web, (2004), 675~684.

Tan, A. and C. Teo, "Learning user profiles for personalized information dissemination", In Proc. Intl. Joint Conf. on Neural Network, 183~188, 1998.

Trajkova, J. and S. Gauch, "Improving ontology-based user profiles", *In Proc. the RIAO*, (2004), 380~389.

Wang, J., Z. Li, J. Yao, Z. Sun, M. Li, and W. Ma, "Adaptive user profile model and collaborative filtering for personalized news", APWeb, (2006), 474~485.

Zayani, C., A. Peninou, M. Canut, and F. Sedes, "An adaptation approach : query enrichment by user profile", In Proc. IEEE/ACM Signal-Image Technology and Internet Based Systems (SITIS), (2006), 24~35.

Abstract

PIRS : Personalized Information Retrieval System using Adaptive User Profiling and Real-time Filtering for Search Results

Hocheol Jeon* · Joongmin Choi*

This paper proposes a system that can serve users with appropriate search results through real time filtering, and implemented adaptive user profiling based personalized information retrieval system(PIRS) using users' implicit feedbacks in order to deal with the problem of existing search systems such as Google or MSN that does not satisfy various user' personal search needs. One of the reasons that existing search systems hard to satisfy various user'personal needs is that it is not easy to recognize users' search intentions because of the uncertainty of search intentions. The uncertainty of search intentions means that users may want to different search results using the same query. For example, when a user inputs "java"query, the user may want to be retrieved "java"results as a computer programming language, a coffee of java, or a island of Indonesia. In other words, this uncertainty is due to ambiguity of search queries. Moreover, if the number of the used words for a query is fewer, this uncertainty will be more increased. Real-time filtering for search results returns only those results that belong to user-selected domain for a given query. Although it looks similar to a general directory search, it is different in that the search is executed for all web documents rather than sites, and each document in the search results is classified into the given domain in real time. By applying information filtering using real time directory classifying technology for search results to personalization, the number of delivering results to users is effectively decreased, and the satisfaction for the results is improved. In this paper, a user preference profile has a hierarchical structure, and consists of domains, used queries, and selected documents. Because the hierarchy structure of user preference profile can apply the context when users perfomed search, the structure is able to deal with the uncertainty of user intentions, when search is carried out, the intention may differ according to the context such as time or place for the same query. Furthermore, this structure is able to more effectively track web documents search behaviors of a user for each domain, and timely recognize the changes of user intentions. An IP address of each device was used to identify

* The Department of Computer Science and Engineering, Hanyang University

each user, and the user preference profile is continuously updated based on the observed user behaviors for search results. Also, we measured user satisfaction for search results by observing the user behaviors for the selected search result. Our proposed system automatically recognizes user preferences by using implicit feedbacks from users such as staying time on the selected search result and the exit condition from the page, and dynamically updates their preferences. Whenever search is performed by a user, our system finds the user preference profile for the given IP address, and if the file is not exist then a new user preference profile is created in the server, otherwise the file is updated with the transmitted information. If the file is not exist in the server, the system provides Google' results to users, and the reflection value is increased/decreased whenever user search. We carried out some experiments to evaluate the performance of adaptive user preference profile technique and real time filtering, and the results are satisfactory. According to our experimental results, participants are satisfied with average 4.7 documents in the top 10 search list by using adaptive user preference profile technique with real time filtering, and this result shows that our method outperforms Google's by 23.2%.

Key Words : Adaptive User Profile, Real-time Filtering, Personalized Information Retrieval

저 자 소 개

전호철

서원대학교 전자계산학과 공학사(1998), 한양대학교 컴퓨터공학과 공학석사(2000), 현재 한양대학교 컴퓨터공학과에서 박사과정 재직 중이며, 주요 관심분야는 지능형 에이전트, 정보검색/정보추출, 인공지능, 상황인지 등이다.

최중민

서울대학교 컴퓨터공학과를 졸업하였고, 1986년에 서울대학교 대학원 컴퓨터공학과에서 석사학위를, 1993년에 미국 State University ofYork at Buffalo에서 컴퓨터학 박사 학위를 각각 취득하였다. 1993년부터 1995년까지 한국전자통신연구원(ETRI)에서 선임연구원으로 재직하였으며, 1995년부터 현재까지 한양대학교 컴퓨터공학과 교수로 재직 중이다. 한국 정보과학회, 정보처리학회, 지능정보시스템학회, 인터넷정보학회, 미국 IEEE, ACM 등의 정회원이며, 관심분야는 웹지능, 텍스트마이닝, 정보검색/정보추출, 인공지능, 지능형 모바일정보시스템 등이다.