기술특집

인터넷 정보검색 시스템의 연구 동향

김 준 태*, 유 견 아**
(*동국대 컴퓨터공학과 교수, **덕성여대 전산학과 교수)

1. 서 론

인터넷의 급속한 성장에 따라 엄청난 양의 정보가 온라인으로 제공되고 있다. 현재 웹(WWW, World Wide Web)에 존재하는 페이지의 수는 약 1억개 이상으로 추정되고 있으며 계속적으로 증가되고 있는 추세이다. 정보의 양이 급격히 증가함에 따라 사용자가 원하는 지식을 찾기 위한 작업은 더욱 더 어려워지고 있다. 간단한 질의에 대하여 서치엔진(search engine)을 통해 검색되는 웹 페이지의 수는 수만개 이상에 이르고 있으며, 필요한 정보를 얻기까지 점차 많은 시간과 노력이 필요하게 되었다.

이러한 정보의 홍수 속에서 사용자로 하여금 원하는 정보를 쉽게 찾아 볼 수 있도록 도와주는 시스템에 대한 연구가 활발하게 진행되고 있다. 인터넷에서의 정보검색 시스템들을 그 형태와 용도에 따라 분류해 보면 로봇에 의해 웹 페이지들을 수집하여 색인을 만들어 제공하는 서치엔진, 다수의 서치엔진에 질의를 던지고 결과를 취합하여 보여주는 메타서치엔진(metasearch engine), 사용자를 대신하여 백그라운드에서 웹 페이지들을 수집하여 저장하는 오프라인 브라우저(off-line browser), 학습을 통하여 사용자의 관심에 부합하는 웹 페이지들을 추천하는 추천 시스템(recommendation system), 전자 우편이나 usenet 뉴스 메시지를 여과하여 전달하는 메일 및 뉴스 필터링 시스템(filtering system), 사용자가 지정한 정보들을 실시간으로 사용자의 컴퓨터로 전달하는 푸시(push) 서비스 등으로 나누어 볼 수 있다.

이러한 정보검색 시스템들을 구현하기 위해서 다양한 정보에 접근하는 방법, 웹 페이지나 뉴스 메시지 등 수집된 정보를 분석하는 방법, 이들을 사용자의 관심에 따라 여과하고 분류하는 방법, 소프트웨어 에이전트에 의한 정보 교환 방법, 사용자의 반응에 따라 학습을 수행하는 방법 등에 대한 많은 연구가 이루어져 왔으며, 최근에는 이러한 기술들을 응용한 상업화된 서비스나 상용 프로그램들이 등장하고 있는 추세이다.

본 논문에서는 이러한 인터넷 정보검색의 연구 동향을 살펴보고, 검색 시스템을 구현하기 위한 기술들과 최근에 소개된 연구 사례 및 프로그램들을 형태와 용도에 따라 분류하여 설명한다. 2장에서는 인터넷 정보검색과 관련된 기술들을 간략히 설명하고, 3장에서는 다양한 인터넷 정보검색 시스템들을 용도와 형태에 따라 분류하여 소개하며, 4장에서 인터넷 정보검색에 대한 향후 연구 전망에 대해 논한다.

2. 인터넷 정보검색 관련 기술

다양한 형태의 정보가 존재하는 인터넷에서 편리하고도 정확하게 정보를 찾기 위해서는 지능적인 검색 기술이 필요하다. 본 장에서는 지능적인 정보검색을 위해 많이 사용되어지고 있는 기초 기술들로 문서의 색인 방법, 정보의 여과 방법, 협동에 의한 정보 추천, 소프트웨어 에이전트, 그리고 사용자 기호의 학습 방법 등에 대해 설명한다.

2.1 문서 색인

현재 정보검색의 대부분은 검색하려는 웹 페이지나 뉴스 메시지 등의 내용을 나타내는 텍스트를 비교 분석하여 이루어진다. 텍스트를 비교하기 위해서는 우선 텍스트를 이루고 있는 단어들을 추출하여 통계적인 데이터를 산출해야 하는데 이 과정을 색인(indexing)이라고 한다.

문서에서 단어들을 추출하기 위해서는 사전을 사용하여 영문의 경우에는 간단한 stemming을, 국문의 경우에는 형태소 분석 과정을 거친 뒤 각 단어에 그 중요도에 따라 가중치를 부여한다[3, 14, 31]. 단어에 가중치를 주는 방법으로는 각 단어의 문서내 출현 빈도(TF, term frequency)를 정규화하여 사용하는 방법이 있고, 유한개의 키워드를 미리 선정하여 문서를 표현하는 경우에는 표본 문서 집단으로부터 역문헌빈도(IDF, inverted term frequency)를 계산하여 출현 빈도와 함께 사용하기도 한다. 이 경우 문서 j에 있는 단어 i의 가중치는 다음의 식으로 표현된다.

$$W_{ij} = freq_{ij} * (\log \frac{N}{DF_i} + 1)$$

여기서 $freq_{ij}$는 단어 i의 문서 j에서의 출현 빈도, N 은 총

표본 문서의 개수, DF_i는 단어 i를 포함하는 문서의 개수이다. 역문헌빈도의 의미는 모든 문서에 고르게 출현하는 단어는 검색을 위한 색인어로서의 가치가 낮으므로 낮은 가중치를 부여하는 것이다.

텍스트에서 단어들을 추출하는 작업 외에도 문서의 구조에 따라 각 문서를 특징지을 수 있는 요소들을 추출하는 과정이 필요하다. 예를 들어 전자우편의 경우 Sender, Subject 등의 필드를 인식하거나, 웹 페이지의 경우에 HTML tag를 분리해 내는 작업을 수행하여야 한다.

2.2 내용 기반 문서 여과

사용자가 원하는 정보를 선택적으로 제공하기 위해 가장 일반적으로 쓰이는 것은 텍스트를 통계적으로 분석하는 내용 기반 여과(content-based filtering) 방법이다[7, 8, 14, 29, 31]. 새로운 문서(email message, news article 또는 web page)가 사용자의 관심(user interest)에 부합하는가를 판별하는 것은 사용자의 관심과 일치하는 문서의 카테고리를 YES, 일치하지 않는 카테고리를 NO라고 했을 때 두 개의 카테고리를 갖는 분류 문제가 된다.

내용 기반 방법은 사용자의 관심과 일치하는 문서들과 일치하지 않는 문서들의 샘플로부터 단어들을 추출하여 사용자의 관심을 학습하고, 선택 대상 문서를 단어들과 각 단어의 가중치들로 나타낸 뒤 학습된 사용자 관심을 이용하여 YES, 또는 NO로 분류하는 것이다.

내용 기반 여과 방법으로 가장 대표적인 것은 벡터 유사도(vector similarity)를 이용한 방법이다. 단어들을 추출하고 가중치를 계산하면 사용자의 관심과 선택 대상 문서를 모두 단어 가중치의 벡터로 표현할 수 있고, 이들 사이의 유사도를 두 벡터 사이의 각도를 나타내는 cosine similarity를 이용하여 계산한다. 문서의 표현에 총 n개의 단어가 사용되고, 사용자의 관심을 나타내는 단어 가중치 벡터를 P(profile), 선택 대상 문서를 나타내는 단어 가중치 벡터를 D(document)라고 하면 이 두 벡터의 유사도는 다음과 같다.

$$Similarity(P, D) = \frac{P \cdot D}{|P||D|}$$

사용자의 관심과 일치한다고 알려진 문서들의 집합을 S^+, 일치하지 않는다고 알려진 문서들의 집합을 S^-라고 하자. 벡터유사도를 이용한 여과는 S^+의 평균 벡터로 사용자의 관심 P를 나타내고, 선택 대상 문서 D와 P와의 유사도가 일정한 임계값(threshold) T를 넘으면 D를 사용자의 관심에 부합하는 것으로 판단하는 것이다.

내용 기반으로 문서를 여과하는 또 다른 방법은 베이지안 확률(Bayesian probability)을 이용하는 것이다. 이 방법은 S^+와 S^-로부터 확률 $P(Y)$, $P(N)$, $P(W_i|Y)$, $P(W_i|N)$ 등의 값을 추정한 뒤 단어 $W_1, W_2, \ldots, W_n$이 나타난 문서 D가 Y(YES) 또는 N(NO)으로 분류될 확률을 다음과 같이 비교하여 확률값이 높은 쪽으로 D를 분류한다.

$$P(Y \mid W_1, \cdots, W_n) = k * P(Y) * P(W_1|Y) * \cdots * P(W_n|Y)$$
$$P(N \mid W_1, \cdots, W_n) = k * P(N) * P(W_1|N) * \cdots * P(W_n|N)$$

유한개의 키워드와 기타 문서의 특성을 나타내는 요소들로 N개의 특징(feature)을 정의하여 문서를 표현하는 경우, 문서의 여과에 사례 기반 추론(case-based reasoning 또는 memory- based reasoning), 결정 트리(decision tree), 신경망(neural network) 등을 사용할 수도 있다.

사례 기반 추론 방법은 D와 S^+, S^-의 모든 문서들과의 유사도를 계산하여 가장 유사한 문서(nearest neighbor)가 S^+에 속한 것이면 D도 사용자의 기호에 맞는 것으로 판단한다. 결정 트리를 이용하는 방법은 S^+와 S^-의 문서들을 training set으로 하여 사용자의 관심을 판별하는 트리를 학습하고 학습된 결정 트리를 이용하여 새로운 선택 대상 문서를 YES 혹은 NO로 판별한다. 신경망을 이용하는 경우에는 N개의 feature가 input layer, YES/NO가 output layer를 형성하는 multi-layered backpropagation network를 구성하고 S^+와 S^-의 문서들을 training set으로 하여 학습을 수행한다.

2.3 협동에 의한 정보 추천

협동에 의한 정보 추천(collaborative recommendation)은 대상 문서들의 내용을 비교하는 대신에 사용자들 사이의 기호를 비교하는 것이다. 즉, 사용자가 이전에 선호한 정보를 토대로 새로운 아이템을 추천하는 것이 아니라 유사한 취향의 다른 사용자가 선호하는 아이템을 추천하는 방식이다[13, 20, 30, 36, 46].

예를 들면, 음반을 추천하는 경우 각 사용자는 몇가지 음반에 대하여 좋아한다, 싫어한다, 보통이다 등의 등급을 지정한다. 추천 시스템은 다양한 음반에 대한 각 사용자의 응답을 이용하여 사용자들을 클러스터링(clustering)한다. 만일 사용자 A와 B가 같은 클러스터에 있고 A는 음반 1, 2, 3, 4에 대하여 좋다는 응답을 했으며, B는 1, 2, 3, 5에 대하여 좋다고 했다면, 사용자 A에게는 음반 5를, 사용자 B에게는 음반 4를 추천하는 것이다.

이를 위해 비슷한 취향의 사용자들을 그룹화 하는 작업이 필요하다. 기호가 같은 사용자를 찾는 방법으로는 각 사용자의 특성을 추천하려는 정보의 여러 속성으로 표현하고 nearest neighbor를 계산하는 방법, 확률적으로 사용자를 클러스터링 하는 방법, 에이전트(agent) 사이의 정보 교환에 의해 같은 기호의 사용자 정보를 점진적으로 획득하는 방법 등이 사용되고 있다.

협동에 의한 정보 추천은 인간이 판단한 정보를 직접 이용하기 때문에 비문자 정보를 추천할 수 있다는 상대적 장점이 있지만, 그룹 내의 어떤 사용자도 보지 못한 완전히 새로운 아이템이 나타났을 때 추천이 불가능하며, 독특한 기호를 지닌 사용자를 수용할 적절한 방법이 없다는 단점이 있다.

2.4 다중 에이전트에 의한 정보 검색

사용자를 대신하여 작업을 수행하는 소프트웨어 에이전트

(software agent)에 대한 연구는 최근 들어 많은 관심이 집중되는 분야이며, 인터넷에서의 정보검색에 에이전트 기술을 활용하기 위한 연구도 활발하게 이루어지고 있다[10, 15, 17].

에이전트의 속성, 에이전트의 구현에 이용되는 기술, 에이전트의 구조, 에이전트의 응용분야 등은 매우 다양하기 때문에 에이전트를 한마디로 정의하긴 어렵지만 많은 연구자들은 에이전트가 갖추어야 할 속성들에 대해 공통적인 성질들을 기술하고 있다. 에이전트의 대표적인 성질로는 사용자의 간섭을 받지 않고 사용자로부터 위임받은 일을 자율적으로 처리하는 자율성(autonomy), 다른 에이전트나 사람과 상호작용을 하는 사교성(social activity), 추론과 학습 능력을 갖추어 새로운 지식을 습득하는 지능 (intelligence), 자신이 추구하는 목적을 달성하기 위한 능동적인 행동을 하는 선행위성(proactiveness), 사용자의 습관이나 작업 방식 및 취향에 따라 스스로를 변화시키는 적응성(adaptivity), 네트워크 상에서 이동할 수 있는 능력인 이동성 (mobility) 등이 있다[5, 9].

인터넷에서와 같이 분산되어 있고(distributed) 이질적인(heterogeneous) 정보 소스들로부터 다양한 매체의 자료를 검색하기 위해서 이러한 에이전트들이 각자의 목적에 따라 네트워크의 여러 곳을 돌아다니며 다른 에이전트와 협력하여 정보를 교환할 수 있는 다중 에이전트 환경이 연구되고 있다. 그림 1은 이러한 다중 에이전트 시스템의 개념을 보여주고 있다.

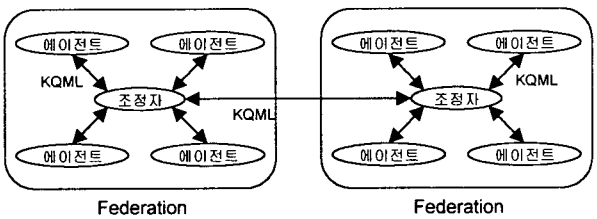

그림 1. 다중 에이전트 시스템

다중 에이전트 환경에서 에이전트는 조정자(facilitator, mediator, broker)라고 불리는 별도의 에이전트를 통해 다른 에이전트들과 통신한다. 하나의 조정자와 그에 등록된 에이전트들은 한 연방(federation)을 이루어 모든 에이전트들은 통신에 관한 자율성을 모두 그들의 조정자에게 위임하며 조정자는 그들을 관할한다. 다른 연방에 속한 에이전트와의 통신 또한 이들 조정자들을 통해서 이루어진다.

조정자는 에이전트 생성시 에이전트의 식별자와 기능을 등록 받고 에이전트로부터 통신 메시지를 받으면 그 요구에 적합한 에이전트를 찾아 메시지를 전달한다. 필요에 따라 조정자는 메시지를 분할하여 여러 에이전트로 보내기도 하고 여러 메시지를 합성하기도 한다. 조정자라는 추가적인 에이전트의 이용은 에이전트끼리 직접 통신할 때 모든 에이전트가 다른 에이전트에 대한 정보를 갖고 있어야 하며 통신 협상도 직접 해야 하는 등의 부하를 감소시킬 수 있다.

이질적인 에이전트 사이의 통신을 위해서는 표준화된 에이전트 통신언어가 필요한데 ARPA의 KSE (Knowledge Sharing Efforts)에서 제안한 KQML (Knowledge Query and Manipulation Language)이 널리 이용되고 있다 [12].

2.5 사용자 관심 학습

정보검색 시스템에서의 학습이란 명시적, 혹은 묵시적으로 획득된 사용자의 피드백(user feedback)으로부터 사용자의 관심을 나타내는 프로파일(profile)을 갱신하여 사용자의 관심을 점차 정확하게 표현하는 것이다. 사용자의 관심을 학습하는 방식은 사용자로부터 피드백을 얻는 방식과 사용자 프로파일을 변경하는 알고리즘에 따라 구분할 수 있다.

사용자로부터 피드백을 얻는 방식은 크게 명시적(explicit) 방법과 암시적(implicit) 방법으로 나누어 볼 수 있다. 명시적 피드백이란 사용자가 시스템이 제시한 특정 문서에 대해 직접 자신의 관심과 일치하는지 여부를 입력하는 방식으로, Yes/No의 두 단계 혹은 그 이상의 단계로 등급을 입력하는 것이다[26, 8, 29]. 이에 반하여 암시적 피드백은 시스템이 사용자의 행위를 관찰하여 특정 문서에 대한 사용자의 관심 여부를 판별하는 것으로, 사용자의 파일 삭제 및 이동 행위, 사용자가 특정 문서를 읽는데 소비한 시간, 또는 웹 페이지에서 특정 링크를 클릭하는 행위 등을 바탕으로 사용자의 관심 여부를 추정한다[7, 25].

사용자 프로파일을 변경하는 학습 알고리즘은 문서 여과 방식과 직접 관련이 있다. 일반적으로 많이 이용되는 벡터유사도를 사용하는 경우 적합성 피드백(relevance feedback)에 의한 단어 가중치 변경 방법이 사용된다[32]. 적합성 피드백에 의한 사용자 프로파일 학습은 사용자가 관심과 일치한다고 반응한 문서에 있는 단어들의 가중치는 높이고, 일치하지 않는다고 반응한 문서에 있는 단어들의 가중치는 낮추어 프로파일의 내용을 변경하는 것이다. 현재의 사용자 프로파일 벡터를 P, 사용자의 기호와 일치하는 문서들의 벡터를 D^+, 사용자의 기호와 일치하지 않는 문서들의 벡터를 D^- 라고 하면, 학습에 의해 변경되는 새로운 프로파일 P'는 다음과 같이 계산된다.

$$P' = P + \frac{\alpha}{N}\sum D^+ - \frac{\beta}{M}\sum D^-$$

위의 식에서 N과 M은 각각 D^+와 D^-의 개수이며, α와 β는 적합한 문서와 부적합한 문서에 대한 학습 가중치로, 이 값들의 조정함으로써 새로운 피드백에 대하여 학습하는 성향을 조정 할 수 있다.

문서 여과에 결정 트리를 이용하는 경우 ID3 알고리즘, 신경망을 사용하는 경우 backpropagation 알고리즘 등이 학습에 사용되며, 사례 기반 추론이나 협동적 여과 방식을 사용하는 경우는 새로운 사례가 추가되는 것이 학습의 역할을 한다고 볼 수 있다.

3. 인터넷 정보검색 시스템의 종류

본 장에서는 인터넷에서의 정보검색을 위해 개발된 시스템들을 그 형태와 용도에 따라 서치엔진, 메타서치엔진, 오프라인 브라우저, 추천 시스템, 메일 및 뉴스 필터링 시스템, 푸시 서비스 등으로 나누어 설명한다.

3.1 서치엔진

서치엔진(search engine)은 웹에서의 정보검색을 위해 가장 먼저 시도되었으며 현재에도 가장 보편적으로 사용되고 있는 시스템으로 기본적인 구조는 그림 2와 같다. 로봇(robot, 또는 spider, wanderer, crawler, warm 등)은 서치엔진에서 웹 페이지들을 모아오는 역할을 하는 프로그램을 말한다. 로봇은 인터넷에 있는 각각의 웹 서버(web server)에 직접 접근하여 웹 페이지들을 검색엔진으로 가져오며, 모아온 페이지들로부터 색인어들을 추출하여 인덱스 데이터베이스를 구축한다. 사용자는 웹 브라우저를 통하여 서치엔진에 접속하고, 단어의 집합 형태로 질의를 던지면 서치엔진은 인덱스 데이터베이스를 참조하여 해당하는 URL들의 리스트를 보여준다.

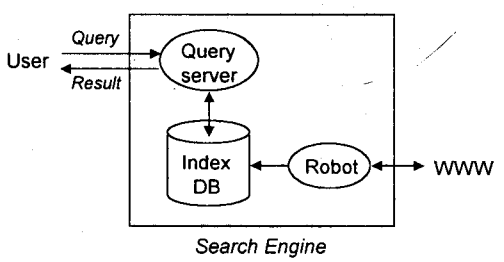

그림 2. 서치엔진

로봇의 기본적인 형태는 특정한 웹 사이트들로부터 시작하여 하이퍼링크를 따라가면서 새로운 웹 페이지들을 가져오는 것으로, 하나의 웹 페이지를 가져오면 그 페이지 내에 있는 모든 URL들을 queue에 저장하고 다음번 URL을 queue에서 선택하여 웹 페이지를 가져오는 작업을 반복한다. WebCrawler[55]는 초기 URL에서부터 출발하여 단순한 너비우선(breadth-first)으로 탐색을 진행하며, Lycos spider[49]와 같은 로봇은 많이 참조되는 URL을 우선 탐색하는 방식을 취하고 있다.

색인 대상으로는 웹 페이지의 타이틀이나 처음 일부 텍스트만을 이용하는 경우도 있고, 웹 페이지 안의 전체 텍스트를 색인하는 경우도 있으며, 색인어에 복잡한 통계적 가중치를 주는 시스템들도 있다. 대표적인 서치엔진에는 AltaVista[41], Yahoo[59], Excite[45], HotBot[47], Lycos[49] 등이 있다.

서치엔진의 공통적인 문제점은 단순 질의에 대한 검색 결과가 지나치게 많고 정확도가 떨어진다는 것과, 로봇에 의한 색인은 텍스트에 국한되어 있고, 이미지와 같은 정보는 검색하지 못한다는 것이다.

3.2 메타서치엔진

메타서치엔진(metasearch engine)은 서치엔진의 로봇과 달리 직접 웹 서버들에 접근해서 웹 페이지들을 모아오지 않고, 사용자가 질의를 주면 다수의 검색엔진에 질의를 던져 그 결과를 모아서 정리하여 사용자에게 제시하는 시스템으로, 대표적인 메타서치엔진으로는 MetaCrawler[50], SavvySearch[52], All4one [40], 그리고 국내 사이트로 미스다찾니[37], 정보탐정[38] 등이 있다. 그림 3은 메타서치엔진의 개념을 나타낸다.

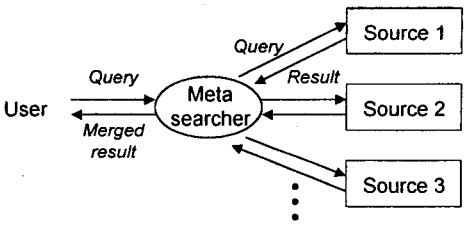

그림 3. 메타서치엔진

메타서치를 하기 위해서는 질의를 던질 대상 서치엔진의 URL, 각 서치엔진의 질의 형태, 검색 결과의 양식 등을 알고 있어야 하며, 각 서치엔진의 검색 결과를 취합하여 중복을 제거하고 유사도에 따라 순위를 정하는 알고리즘이 정의되어야 한다.

웹 페이지를 검색하기 위해 서치엔진에 질의를 던지는 것과 비슷하게 쇼핑을 위한 상품 정보를 검색하기 위해 다수의 인터넷 쇼핑몰에 접속하여 정보를 찾아오는 시스템들도 있다. Jango[48]는 쇼핑 정보를 검색하는 사이트들로서, 사용자가 검색하고자 하는 상품의 종류와 사양을 선택하면 다수의 인터넷 쇼핑몰로부터 관련 상품의 정보를 찾아 종합하여 보여준다.

3.3 오프라인 브라우저

메타서치엔진과 비슷한 역할을 하는 독립적인 프로그램으로 사용자를 대신하여 백그라운드에서 사용자의 관심에 맞는 정보를 수집하여 저장하는 프로그램을 오프라인 브라우저(off-line browser) 또는 정보 수집 에이전트(information gathering agent)라고 하며, WebZip[57], WebCompass[54], WebTamer [56], Agentware [39] 등 다수의 상용화된 프로그램들이 있다.

WebZip은 사용자가 지정하는 웹 사이트에서 지정된 깊이까지 모든 웹 페이지를 수집하여 사용자의 하드디스크에 저장하는 역할을 한다. 사용자는 작업을 지시해 놓고 작업이 끝난 후 오프라인으로 빠르게 수집된 웹 페이지들을 검색할 수 있다. WebCompass, WebTamer 등은 사용자의 질의를 받아 다수의 서치엔진과 ftp 사이트, Gopher 사이트 등에 접속하여 정보를 가져오고 결과를 취합하여 제공한다. 이들은 단순한 정보 수집 외에도 웹 사이트의 변화를 알려주는 등의 다양한 기능을 제공한다. Agentware는 사용자의 자연언어 질의를 분석하여 패턴 매칭과 신경망에 의해 수집된 정보를 여과하며, 사용자의 행위를 관찰하여 학습을 수행한다.

Shopping Explorer[53]는 오프라인으로 상품 정보를 찾아오는 쇼핑 에이전트로서 사용자가 검색하고자 하는 상품 정보에 대해 다수의 인터넷 쇼핑몰에 접속하여 정보를 수집해 제공하는 역할을 한다.

3.4 추천 시스템

추천 시스템(recommendation system)은 지능형 에이전트로서 사용자의 관심 분야를 스스로 파악하여 학습하고, 사용자의 관심 분야에 부합하는 정보들을 수집하여 제공한다. 대

부분 학습을 위하여 사용자로부터 추천된 정보에 대한 관련 여부를 피드백(feedback)으로 받는다. 정보 추천 에이전트는 추천 방식에 따라 브라우징 어시스턴트(browsing assistant), 내용기반 추천 시스템(content-based recommendation system), 협동적 추천 시스템(collaborative recommendation system) 등이 있다.

브라우징 어시스턴트는 사용자가 웹 페이지를 읽고 있는 동안 그 페이지 안의 링크들 중 사용자의 관심과 부합하는 링크를 추천하는 시스템으로, WebWatcher[7], Letizia[25] 등이 있다. WebWatcher는 카네기멜론 대학에서 만든 에이전트로, 사용자가 링크를 선택하는 행위를 관찰하여 사용자의 관심 분야를 학습한다. 사용자가 선택한 링크들을 학습 예제로 하여 단어들과 각 단어에 대한 가중치로 사용자의 관심 분야를 나타내고 새로운 링크들에 대하여 벡터유사도에 의해 사용자의 관심과 부합 여부를 결정한다. Letizia는 웹 브라우저 위에서 동작하는 프로그램으로, 사용자가 보고있는 페이지의 링크들을 미리 가져와서 평가하고, 사용자가 도움을 요청할 경우 링크를 추천한다.

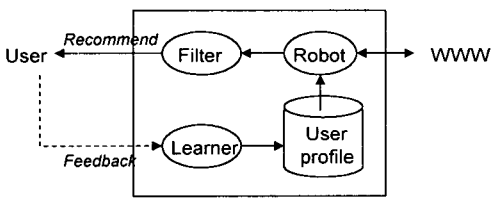

그림 4. 내용기반 추천 시스템

내용기반 추천 시스템은 특정한 웹 페이지에 대한 사용자의 평가를 기초로 하여 사용자의 관심을 학습하고 새로운 웹 페이지들을 수집하여 제공하는 것으로(그림 4), Syskill&Webert [29], Fab[8], Amalthaea[28] 등의 시스템이 있다.

Syskill&Webert는 기본적으로 제공하는 인덱스 페이지의 일부 링크에 대한 사용자의 평가를 기초로 사용자의 관심을 학습한다. 학습을 위해 사용자가 평가한 페이지들로부터 특징 벡터를 추출하여 Bayesian classification, nearest neighbor, decision tree, neural network 등 다양한 방법으로 학습을 수행하고 그 결과에 따라 웹 페이지를 추천한다. 사용자 관심에 부합하는 링크를 추천하는 방식은 인덱스 페이지의 링크 중 나머지들을 추천하는 방식과, 사용자 관심을 질의로 표현하여 Lycos 서치엔진에 질의를 던져 새로운 웹 페이지를 수집하는 두 가지 방식으로 동작한다. Fab은 다수의 사용자들을 관심 분야별로 그룹화 하여 내용기반 추천과 협동적 추천을 복합적으로 사용한다. Fab은 주제별로 웹 페이지를 모아오는 메타서치 형태의 collection agent와 사용자별로 웹 페이지를 제공하는 selection agent로 구성되어 있으며, 사용자별로 관심을 나타내는 프로파일을 구성하고 추천된 페이지에 대한 사용자의 평가에 따라 프로파일에 있는 단어들의 가중치를 수정함으로서 학습을 수행한다. 내용기반 추천은 벡터유사도를 기반으로 하며, 협동적 추천은 한 사용자가 높은 평가를 한 웹 페이지를 같은 그룹에 있는 다른 사용자들에게도 추천함으로써 이루어진다. Amathaea는 다수의 에이전트가 웹 페이지의 수집과 여과를 수행하며 유전자 알고리즘을 적용하여 추천 정확도가 높은 우수한 에이전트가 진화 과정을 거쳐 발전하도록 하였다.

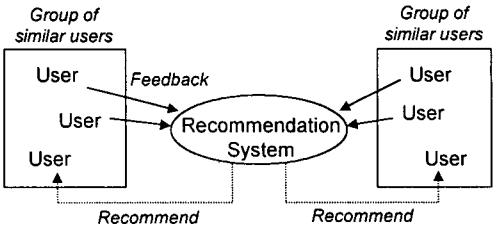

그림 5. 협동적 추천 시스템

협동적 추천 시스템은 웹 페이지나 뉴스 메시지 등의 텍스트를 분석하지 않고, 다수의 사용자를 그룹화하여 일부 사용자들로부터 높은 평가를 받은 정보를 다른 사용자들에게도 추천하는 것으로, 내용기반 추천이 어려운 음악, 영화 등의 추천에 적합하다(그림 5). Siteseer[30]와 같은 시스템은 사용자들의 북마크에 있는 URL만을 이용하여 유사한 북마크 폴더들을 찾고, 유사한 북마크를 가진 사용자들 사이에 상호 URL을 추천한다. Phoaks[36]는 USENET의 각 뉴스그룹에 관련 URL을 추천하는 시스템으로, 웹 사이트를 추천하는 내용의 뉴스 메시지들을 인지하여 메시지 안에 언급된 URL들을 수집하고 언급된 회수가 많을수록 관련 있는 웹 페이지로 간주하여 추천한다. FireFly[46]는 대표적인 협동적 추천 시스템으로, 영화나 음반에 대한 사용자들의 평가를 기초로 비슷한 취향의 사용자 그룹을 만들고, 같은 그룹 내의 사용자들로부터 높은 평가를 받은 영화나 음반을 새로운 사용자에게 추천한다. WiseWire[58]는 영화나 음반에 대한 정보뿐만 아니라 웹 페이지, 뉴스 메시지, ftp 사이트 등 다양한 소스로부터의 정보를 협동적 방식과 내용기반 방식을 복합적으로 사용하여 추천하는 시스템으로, 사용자의 평가를 기초로 한 신경망 학습을 사용한다.

3.5 뉴스 및 메일 여과 시스템

메일 여과(mail filtering) 시스템은 사용자가 전자우편을 읽고, 저장하고, 삭제하는 행위를 학습하여 사용자를 대신해 불필요한 메일을 자동으로 삭제하거나, 긴급한 메일의 도착을 알리거나, 또는 메일들을 분류하는 지능형 에이전트로서, Maxim[26], Information Lens[27] 등의 프로토타입 시스템이 있고, Netscape에서는 학습의 기능은 없으나 전자우편을 자동으로 여과하는 필터를 지정할 수 있다. Maxim은 각 email 메시지에 대한 사용자의 행위를 관찰하여 결과를 [situation, action] 쌍으로 기억한다. 새로운 메일이 도착하면 메시지의 특징들을 추출하고(Sender, Subject 등) 과거의 사례와 비교하여 nearest neighbor를 찾은 뒤 그 사례의 action에 따라 사용자의 행위를 예측한다. 이때 현재의 상황과 과거의 사례와의 유사도를 confidence factor로 하여 confidence factor가 아주

높으면 바로 삭제 등의 행동을 취하고 중간 범위에 해당하면 사용자에게 행동을 제안하도록 되어 있다. 학습은 반복되는 사용자의 행위를 사례로 계속 추가함으로써 이루어진다. Information Lens나 Netscape mail은 사용자가 입력하는 명시적인 룰에 의해 메일을 여과하고 분류하는 시스템으로 학습의 기능은 없으며 메일의 각 필드의 텍스트에 스트링 매칭을 하여 룰을 적용시킨다.

뉴스 여과(news filtering) 시스템은 수만개에 달하는 USENET 뉴스 그룹과 매일 수백개 이상씩 올려지는 각 뉴스 그룹의 메시지들로부터 사용자의 관심에 부합되는 메시지들만을 선별하여 제공하는 시스템을 말한다. GroupLens[20]와 같은 시스템은 협동적 방법을 사용하여 각 메시지에 대한 뉴스그룹 이용자들로부터의 평가를 기초로 높은 평가를 받은 메시지들만을 여과하는 시스템이다. NewT[26]는 내용기반 시스템으로, 사용자는 우선 적합한 뉴스 메시지와 부적합한 뉴스 메시지의 예들을 제공하여 시스템을 초기화한다. 이후 NewT는 초기화된 정보를 바탕으로 계속적으로 뉴스 메시지를 여과하여 전달하고, 사용자는 NewT에 의해 여과된 메시지들에 대해 적합 또는 부적합 판정을 내린다. NewT에 의한 메시지의 여과는 단어들을 기반으로 한 벡터유사도를 사용하며, 사용자의 반응에 따라 프로파일의 가중치를 변경함으로써 학습을 수행하여 사용자와의 상호작용이 계속됨에 따라 여과의 정확도가 향상된다.

3.6 푸시 서비스

사용자가 브라우저를 통해 필요한 정보를 서버로부터 가져오는 기존의 방식을 정보를 끌어온다는 의미에서 풀(pull) 모델이라고 할 때, 이에 반해 정보를 갖고 있는 서버 측에서 정보를 밀어 주는 형식을 푸시(push) 모델이라고 한다[4, 23, 24]. 푸시는 netcasting, webcasting, personal information delivery 등 다양한 용어로 표현되고 있다. 푸시서비스는 1996년 PointCast[51]가 뉴스서비스를 무료로 전파하면서 본격적으로 소개되었지만 그 이전부터 사용하던 브라우저의 What's New 나 What's Cool과 같은 기능은 서버가 사용자에게 능동적으로 정보를 제공했다는 점에서 푸시의 일종이라고 할 수 있다. 일반적으로 푸시서비스를 이용하려면 우선 푸시서비스를 제공하는 서버로 가서 사용자 등록을 한 후 거기서 제공하는 푸시 클라이언트 프로그램을 다운로드 받아 설치해야 한다. 푸시 클라이언트 프로그램을 설치하고 나면 사용자는 자신이 원하는 정보의 종류와 그 정보들을 가져오는 주기 등의 환경설정을 할 수 있다. 푸시 클라이언트는 사용자의 환경 설정에 따라 정해진 시간에 서버에 접속하여 지정된 정보를 받아 놓으면 사용자는 자신이 편리한 시간에 저장된 정보를 볼 수 있다 (그림 6).

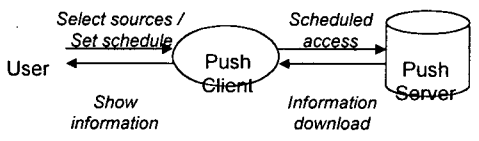

그림 6. 푸시 서비스

푸시서비스를 이용하는 순서를 자세히 보면 정보 전달 방식이 클라이언트가 서버에 요청하여 클라이언트가 정보를 끌어오는 형태로서, 엄밀한 의미에서는 정해진 시간마다 자동으로 클라이언트가 서버로부터 정보를 끌어오는 자동화된 풀(automated pull) 혹은 폴링 푸시(polling push)라고 할 수 있다. 이 과정은 사용자에게는 드러나지 않게 진행되어 사용자 입장에서는 서버가 자신에게 필요한 정보를 푸시해준 것으로 인식되는 것이다.

현재 이용되고 있는 많은 푸시 제품들이 위와 같은 방식으로 동작하는데 서비스하는 정보의 종류나 정보의 전달 방식 등에서 차별화된다. 푸시서비스의 선두주자인 PCN의 PointCast는 각종 뉴스정보(CNN, whether, sports, etc.)를 PointCast 서버에 통합하여 저장해 놓고 사용자가 선택한 뉴스들을 제공하며 클라이언트에서는 스크린 세이버(screen saver)로 갱신된 뉴스들을 보여준다. 반면에 Marimba의 Castanet[44]은 뉴스보다는 소프트웨어나 응용프로그램 등 프로그램 단위의 정보를 다룬다. 사용자는 Castanet을 통해 소프트웨어 벤더(vendor) 등의 원하는 정보원에 등록을 하고 채널을 확보하면 이 채널을 통해 Castanet 클라이언트들은 정보원에 접근할 수 있다. BackWeb[42]은 사용자가 원하는 정보를 멀티미디어 클립 형태로 정기적으로 제공한다. 특히 플래쉬(flash)와 같은 기능은 사용자가 정상적인 컴퓨터 작업을 하고 있을 때 서버로부터 정보가 왔음을 알리는 작은 윈도우가 화면 하단에 나타나며 이를 마우스로 누르면 전달된 정보를 보여주고 그렇지 않으면 잠시 후 사라진다.

이와 같은 독자적인 푸시 제품들 이외에 마이크로소프트나 넷스케이프는 기존의 브라우저에 내장된 형태로 푸시 기능을 제공하고 있다. 마이크로소프트 인터넷 익스플로러(Internet Explorer) 4.0은 Favorites 메뉴에서 원하는 정보원을 추가해 놓고 그 사이트에 가입만 하면 자동적으로 갱신된 정보를 받아와서 사용자가 오프라인으로 정보를 볼 수 있으며, 넷스케이프 커뮤니케이터(Communicator) 4.0에서는 커뮤니케이터의 한 구성 요소인 넷캐스터(NetCaster)를 통해 위에서 소개된 여러 푸시 제품들과 유사한 서비스를 제공한다.

푸시서비스는 원하는 정보를 백그라운드 작업으로 혹은 오프라인으로 받아 볼 수 있어 사용자가 브라우저 앞에서 기다려야 하는 시간을 절약 해줄 뿐 아니라, 그 정보의 변화를 계속적으로 추적해야 하는 번거로움을 덜어 줄 수 있다. 그러나 푸시는 네트워크의 전송부하를 가중시킨다는 문제점이 있으며, 누구나 능동적으로 원하는 정보원에 접근하도록 하는 개방된 인터넷이 일부의 푸시서비스 제공자에 의해 과거의 TV와 같이 일방적으로 정보가 여과되어 전달되는 방식으로 후퇴하는 것이 아닌가 하는 비판도 있다[23]. 이런 비판들에도 불구하고 푸시서비스는 인터넷을 편리하게 사용하고자 하는 사용자들의 요구에 따라 계속 발전할 전망이다.

4. 결론 및 향후 전망

본 논문에서는 인터넷에서의 정보 검색을 위한 기초 기술들

을 설명하고, 보다 편리하고 정확하게 정보를 수집하고 제공하기 위한 다양한 형태의 검색 시스템들에 대하여 알아보았다.

초기의 단순한 서치엔진에서 출발한 인터넷에서의 정보 검색은 점차 각 사용자의 기호에 맞추어 정보를 찾고 제공하는 서비스로 변화해가고 있으며 이미 이러한 상용 서비스나 상품화된 프로그램들이 나타나고 있다. 앞으로 인터넷에서의 정보검색과 정보 제공 서비스에 대한 연구는 보다 지능적인 검색을 구현하기 위한 기술 개발과, 소프트웨어 에이전트에 의한 정보검색에 대한 연구 등이 주축이 될 것이다.

지능적인 검색을 구현하기 위한 연구로는 사용자의 행동으로부터 사용자의 기호와 관심분야를 학습하여 개인화된 서비스를 제공하는 방법(personalization), 자연언어나 음성에 의한 지능형 사용자 인터페이스(intelligent interface), 이미지와 같은 멀티미디어 정보에 대한 검색 방법(multimedia information retrieval) 등에 대한 연구가 계속될 것이며, 소프트웨어 에이전트에 의한 정보검색에 대해서는 다중 에이전트 시스템의 구조(multi-agent system architecture), 에이전트 사이의 통신 프로토콜(agent communication protocol), 이동 에이전트(mobile agent) 등에 대한 연구가 활발하게 진행될 것으로 예상된다.

참고문헌

[1] 신봉기, 김영환, "웹 에이전트", *정보과학회지*, pp.61-67, 3, 1997.

[2] 이근배, 김동석, 원형석, 박미화, "에이전트 기반 정보검색", *정보과학회지*, pp.32-37, August 1998.

[3] 최기선, "한국어 정보검색", *한국정보과학회 논문지*, 제12권, 제8호, pp.24-32, 1994.

[4] 최원태, 최미순, "푸시 기술의 개념 및 응용 사례에 관한 연구", *국회도서관보*, April 1998.

[5] 최중민, "에이전트의 개요와 연구방향", *정보과학회지*, pp.7-16, March 1997

[6] Maristella Agosti, Alan Smeaton, *Information Retrieval and Hypertext,* Kluwer Academic Publishers, 1996.

[7] Robert Armstrong, Dayne Freitag, Thorsten Joahims, and Tom Mitchell, "WebWatcher: A learning apprentice for the World Wide Web," *Proceedings of the 12th National Conference on Artificial Intelligence*, 1995.

[8] Marko Balabanovic and Yoav Shoham, "Fab: Content-based, collaborative recommendation," *Communications of the ACM*, 40(3): 66--72, March 1997.

[9] Alper Caglayan and Colin Harrison, *Agent Sourcebook*, the Wiley press, 1997.

[10] E. H. Durfee, D. L. Kiskis, and W. P. Birmingham, "The agent architecture of the University of Michigan digital library," *IEE Proceedings Software Engineering*, Vol. 144, No. 1, February, 1977.

[11] Oren Etzioni and Daniel Weld, "A softbot-based interface to the internet," *Communications of the ACM*, July 1994.

[12] Tim Finin, Yannis Labrou, and James Mayfield, "KQML as an agent communication language," in Jeffrey M. Bradshaw (edited), *Software Agents*, The MIT Press, 1997.

[13] Leonard Fonder, "Yenta: A multi-agent, referral-based matchmaking system," *Proceedings of the First International Conference on Autonomous Agents*, 1997.

[14] W. Frakes and R. Baeza-Yates, *Information Retrieval*, Prentice Hall, 1992.

[15] Michael R. Genesereth, "An agent-based framework for interoperability," in Jeffrey M. Bradshaw(edited), *Software Agents*, The MIT Press, 1997.

[16] Michael R. Genesereth and Steven P. Ketchpel, "Software agents," *Communications of the ACM*, 37(7): 48-54, July 1994.

[17] B. Hermans, "Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the information society & a prediction of (near-)future developments," http://www.comedia.com/broadcatch/agent_thesis/

[18] Thorsten Joachims, Dayne Freitag and Tom Mitchell, "WebWatcher: A tour guide for the World Wide Web," *Proceedings of the 15th International Joint Conference on Artificial Intelligence,* 1997.

[19] Henry Kautz, Bart Selman and Mehul Shah, "ReferralWeb: Combining social networks and collaborative filtering," *Communications of the ACM*, March 1997.

[20] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. herlocker, Lee R.Gordon and John Riedl, "GroupLens: Applying collaborative filtering to usenet news," *Communications of the ACM*, March 1997.

[21] Bruce Krulwich and Chad Burkey, "The InfoFinder agent: Learning user interests through heuristics," *IEEE Expert,* 12(5), 1997.

[22] Bruce Krulwich and Chad Burkey, "The ContactFinder agent: Answering bulletin border questions with referrals," *AAAI-96/IAAI-96 Proceedings*, August 4-8, 1996.

[23] Juhnyoung Lee and S. M. Chung, "Information delivery on the internet: Beyond search engines," *SERI Journal*, Vol. 2, No. 1, 1998.

[24] Drew Leonard, "Channel truf: push content stakes out your screen," *CNETreviews*, http://www.cnet.com/Content/Reviews/Compare/Push/, February 1997.

[25] Henry Lieberman, "Letizia: An agent that assists web browsing," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995.

[26] Pattie Maes, "Agents that reduce work and information overload," *Communications of the ACM*, 37(7):30--40, July 1994.

[27] Thomas W. Malone, Kum-Yew Lai, and Kenneth R.

Grant, "Agent for information sharing and coordination: A history and some reflections," in Jeffrey M. Bradshaw (edited), *Software Agents*, The MIT Press, 1997.

[28] Alexandrous Moukas and Giorgos Zacharia, "Evolving a multi-agent information filtering solution in Amalthaea," *Proceedings of the 1st International Conference on Autonomous Agents*, February 1997.

[29] Michael Pazzani, Jack Muramatsu and Diniel Billsus, "Syskill & Webert: Identifying interesting web sites," *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996.

[30] James Ruker and Marcos J. Polanco, "Siteseer: Personalized navigation for the Web," *Communications of the ACM*, 40(3), pp.73-75, March 1997.

[31] Gerald Salton, "Automatic Text Processing", Addison Wesley Publishing Co., 1989.

[32] Gerald Salton and C. Buckely, "Improving retrieval performance by relevance feedback," *Journal of the American Society for Information Science*, 41, 288297, 1990.

[33] Erik Selberg and Oren Etzioni, "Multi-service search and comparison using the MetaCrawler," *Proceedings of the 4th World Wide Web Conference*, 1995.

[34] Jonthan Shakes, Marc Langheinrich and Oren Etzioni, "Dynamic reference sifting: A case study in the homepage domain," *Proceedings of the Sixth International World Wide Web Conference*, pp.189-200, 1997.

[35] B. Sheth and P. Maes, "Evolving agents for personalized information filtering," *Proceedings of the Ninth Conference on Artificial Intelligence for Applications*, pp. 345--352, 1993.

[36] Loren Terveen, Will Hill, Brian Amento, David McDonald, and Josh Creter, "PHOAKS: A system for sharing recommendations," *Communications of the ACM*, march 1997.

[37] 미스다찾니, http://www.mochanni.com

[38] 정보탐정, http://www.infocop.com

[39] Agentware, http://www.autonomy.com/teach

[40] All4One, http://www.all4one.com

[41] AltaVista, http://www.altavista.com

[42] BackWeb, http://www.backweb.com/html

[43] BargainFinder, http://bf.cstar.ac.com/bf

[44] Castanet, http://www.marimba.com/products

[45] Excite, http://www.excite.com

[46] FireFly, http://www.firefly.com

[47] HotBot, http://www.hotbot.com

[48] Jango, http://www.jango.com

[49] Lycos, http://www.lycos.com

[50] MetaCrawler, http://www.metacrawler.com

[51] PointCast, http://www.pointcast.com/products/pcn

[52] SavvySearch, http://www.savvysearch.com

[53] Shopping Explorer, http://www.shoppingexplorer.com

[54] WebCompass, http://www.quarterdeck.com

[55] WebCrawler, http://www.webcrawler.com

[56] WebTamer, http://www.agentsoft.com

[57] WebZip, http://www.spidersoft.com/webzip

[58] WiseWire, http://www.wisewire-corp.com

[59] Yahoo, http://www.yahoo.com

※ 본 연구는 1999년도 동국대학교 논문게재연구비 지원으로 이루어졌음.

김준태(金埈台)

1963년 10월 10일생. 1986년 서울대 제어계측공학과 졸업. 1990년 미국 University of Southern California 전기공학과 졸업(석사). 1995년 미국 University of Southern California 컴퓨터공학과 졸업(공박). 현재 동국대학교 컴퓨터공학과 조교수.

유견아(柳絹娥)

1964년 1월 27일생. 1986년 서울대 제어계측공학과 졸업. 1988년 서울대 제어계측공학과 졸업(석사). 1995년 미국 University of Southern California 전산학과 졸업(공박). 현재 덕성여자대학교 전산학과 전임강사.