

정보추출을 이용한 학습기반의 웹 인터페이스 에이전트

Web Interface Agent based on Learning using Information Extraction

이말레(Mai-rey Lee)*, 배금표(Keum-pyo Bae)**

초 록

사용자는 원하는 자료를 검색하기 위해서 각 위치에 대한 정보를 저장하고 있는 검색엔진을 이용하는 경우가 대부분이다. 하지만 자료의 양이 방대해짐에 따라 사용자에게 실제로 필요한 정보가 아닐 경우가 많이 발생한다. 본 논문에서는 이러한 문제를 해결할 수 있는 개인형 웹 인터페이스 에이전트 시스템인 웹 가이드를 제안하였다. 웹 가이드는 사용자의 행동과 에이전트의 방문을 키워드를 중심으로 각각의 사례로 저장하는 사례기반 학습 방법을 이용, 특정 개인 사용자가 웹 상에서 검색하고자 하는 자료를 입력받은 후부터 사용자의 방문 행동을 학습하여 보다 빠른 시간 내에 원하고자 하는 자료를 검색할 수 있도록 도와주는 에이전트 시스템이다.

ABSTRACT

Users usually search for the required information via search engines which contain locations of the information. However, as the amount of data gets large, the result of the search is often not the information that users actually want. In this paper a web guide is proposed in order to resolve this problem. The web guide uses case-based learning method which stores and utilizes cases based on the keywords of user's action and agent's visit. The proposed agent system learns the user's visiting actions following the input of the data to be searched, and then helps rapid searches of the data wanted.

키워드: 정보검색 에이전트, 웹가이드, 학습기반, 에이전트 시스템, information retrieval agents, web guide, case-based learning, agent system.

* 여수대학교 멀티미디어학부 조교수

** 중앙대학교 문헌정보학과 박사과정수료(korea@sunchon.ac.kr)

■ 논문 접수일 : 2002년 1월 25일

■ 게재 확정일 : 2002년 3월 6일

1 서 론

1.1 연구의 목적

정보가 늘어감에 따라 인터넷을 이용하는 사용자들은 이러한 수많은 정보 중에서 자신이 실제로 원하는 정보를 구하려는 요구가 발생하게 되었다. 하지만 자신이 원하는 정보를 찾기까지에는 상당한 시간과 노력이 투자되어야 한다. 일반적으로 사용자는 보다 쉽게 원하는 자료를 검색하기 위해서 각 위치에 대한 정보를 저장하고 있는 검색엔진을 이용하는 경우가 대부분이다. 하지만 자료의 양이 방대해짐에 따라 검색엔진을 통한 결과물도 사용자들에게는 필요이상의 결과를 보여주는 경우가 빈번하게 발생하여 사용자에게 실제로 필요한 정보가 아닐 경우가 많이 발생한다. 일례로 최근 인터넷 사용자가 가장 많이 사용하는 검색엔진인 “알타비스타(Altavista)”를 이용하여 “Agent”에 관련된 사이트를 검색해보면 약 30만 건이 결과물로 제시된다. 또한 검색엔진은 불특정 다수를 대상으로 하기 때문에 하나의 키워드에 대해서 사용하는 사람에 따라 전혀 다른 결과물을 얻을 수 있다. 이러한 문제를 해결하기 위해 사용자의 행동을 학습하여 사용자가 원하는 정보를 보다 빨리 얻을 수 있도록 도와주는 웹 에이전트에 관한 연구는 이미 활발하게 진행되어 왔다.

Armstrong이 발표한 WebWatcher는

미국의 카네기멜론 대학(Carnegie Mellon University)의 컴퓨터공학과 홈페이지를 방문하는 불특정 사용자를 대상으로 자신이 원하는 정보를 쉽게 검색할 수 있도록 도와주는 에이전트 시스템이다. Web-Watcher는 사용자의 관심키워드를 입력 받은 후 현재 사용자가 검색하고 있는 홈페이지의 링크들 중에서 이미 방문한 경험이 있는 다른 사용자들에 의한 평가를 기준으로 사용자의 관심분야와 관련성이 많은 링크를 추천해 주는 방식을 이용하였다. 사용자는 자신이 방문한 사이트에 대해서 만족도를 평가할 수 있으며 Web-Watcher는 이를 기준으로 다음 사용자에게 추천해주는 방식을 이용한다. Web-Watcher에서는 문서를 표현하는 방식으로 하나의 문서를 단어 단위로 분류하여 Boolean값을 갖는 bag-of-words 방식을 이용하였고 평가함수로는 $TF \cdot IDF$ 를 이용하였다. 그러나 이 연구는 대상자를 불특정 다수로 하여 하나의 키워드에 대해서 사용자 각자가 서로 다른 관점을 가질 수 있으므로 사용자는 추천 받은 링크로부터 자신이 원하는 정보를 얻을 수 없는 경우가 발생할 수 있다(Armstrong 1996).

Dunja는 WebWatcher의 문제점을 보완한 Personal WebWatcher를 제안하였다. Personal WebWatcher는 WebWatcher와는 달리 사용 대상자를 불특정 다수가 아닌 개인 사용자로 하였으며 사용자의 방문 여부에 따라 다음 페이지를 추천하는 방식을 이용하였다. 문서를 표현하는 방식

은 Web-Watcher 와 동일하며 평가함수는 Bayesian Classifier 방식을 이용하였다. 그러나 이 연구는 사용자가 자신의 관심 여부를 입력하는 방식이 아닌 사용자의 행동을 바탕으로 학습하므로 사용자의 관심 여부와 상관없이 관련성이 전혀 없는 사이트도 추천해 주는 경우가 발생할 수 있고 또한 에이전트가 사용자의 관심분야에 대한 가장 적절한 사이트를 추천할 수 있도록 학습하는 기간이 많이 소요되는 문제점이 있다(Dunja 1996).

Marko Balabanovic이 제안한 LIRA는 특정 정보를 검색하는 것이 아닌 일반적으로 서핑하는 것에 중점을 두었다. 사용자의 일반적인 웹 서핑을 기반으로 하여 휴리스틱한 방법으로 특정 페이지들을 추천해 준 후 사용자로부터 결과 값을 평가하여 새로운 페이지를 추천해주는 방식을 이용하였다. LIRA는 문서를 Vector 방식으로 표현하였다. 그러나 이 연구 역시 사용자의 관심여부를 명확히 파악할 수 없기 때문에 사용자에게 비교적 정확한 사이트를 추천하기 위해서는 학습기간이 오래 걸린다는 문제점이 있다(Marko 1996).

본 연구의 목적은 그동안 연구되어진 결과들이 기존의 웹 에이전트가 안고있는 문제점, 즉 사용자가 정보검색시 짧은 시간안에 원하는 적합한 정보를 얻기가 어렵다는 점과 적합한 정보에 접근하기까지 많은 시간이 소요되는 문제점을 해결할 수 있는 개인형 웹 인터페이스 에이전트

시스템인 웹 가이드(Web-Guide)를 제안하는데 있다.

1.2 연구의 내용 및 방법

본 연구는 특정 개인 사용자를 대상으로 하는 인터넷 웹 인터페이스 에이전트인 웹 가이드를 연구하였다.

웹 가이드는 사용자의 행동과 에이전트의 방문을 키워드를 중심으로 각각의 사례로 저장하는 사례기반 학습 방법을 이용하여 특정 개인 사용자가 웹 상에서 검색하고자 하는 자료를 입력받은 후부터 사용자의 방문 행동을 학습하여 보다 빠른 시간 내에 원하고자 하는 자료를 검색할 수 있도록 도와주는 에이전트 시스템이다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 기반연구로 이용된 에이전트와 사례기반 학습에 대하여 설명한다. 3장에서는 본 논문에서 제안한 웹 가이드 시스템에 대하여 설명한다. 웹 가이드의 개요와 전체적인 시스템 구조, 웹 가이드에서 사용하는 문서표현방법, 정보추출방법 등에 대하여 설명하며 제안된 학습 알고리즘인 사례기반 학습방법에 대하여 기술한다. 4장에서는 웹 가이드 시스템을 개발한 개발환경과 실험환경을 설명하며 실험을 통해 나온 결과를 평가한다. 5장에서는 본 논문에 대한 결론을 맺으며 향후 연구과제에 대해 소개한다.

이 연구에서 제안되는 웹 가이드 연구

방법은 일반적인 HTML문서 표현방식에서 하나의 문서에 나타난 단어를 Vector 방식으로 표현하는 TF·IDF-Vector 방식에 태그를 추가하여 검사하는 방식을 이용하였다. 문서의 가중치를 평가하는 정보추출 방식으로는 사용자가 입력한 키워드가 문서에 나타나는 빈도 수를 평가하는 방식과 키워드에 이용된 태그에 가중치를 부여하여 평가하는 방식을 이용하였다. 학습 방법으로는 평가함수를 통해 평가된 사례를 기반으로 한 사례기반 학습을 이용하였다.

2 기반연구

2장에서는 본 논문의 기반연구로 사용된 에이전트와 사례기반 학습에 대하여 설명한다.

에이전트란 인간이 처리해야 할 일련의 작업들을 컴퓨터가 대신하여 처리하는 시스템을 포괄적으로 일컫는다. 이러한 에이전트에는 로봇과 같이 실세계에서 행동을 하는 하드웨어 에이전트와 전자메일의 도착여부를 체크해주는 프로그램과 같은 소프트웨어 에이전트로 분류할 수 있다. 본 논문에서 언급되는 에이전트는 소프트웨어 에이전트로 한정한다. 이러한 에이전트는 일반적으로 다음과 같은 특징을 지니고 있다. 먼저 자율성이 있는데 이는 에이전트가 다른 프로그램과 구별할 수 있는 가장 다른 특징으로 사용자의 특별한 지시가 없어도 에이전트는 스스로 판

단하여 행동하는 특징을 말한다. 두 번째는 학습성이 있는데 학습성은 에이전트 스스로가 특정 문제에 대한 해를 파악하여 자신의 능력을 강화시키는 특징을 말한다. 학습을 위한 알고리즘은 이미 인공지능에서 많이 연구된 기법을 기반으로 수행된다. 세 번째는 이동성인데 이동성은 사용자의 요구를 요구한 호스트에서 수행하지 않고 해당 호스트로 직접 이동하여 수행하는 특성을 말한다. 이러한 이동성은 기존의 클라이언트/서버 개념과 상이한 개념으로 해당 에이전트를 서버에 전송하여 해당 작업을 수행한다. 인터넷의 보급으로 인해 네트워크의 사용량이 증가하면서 중요성이 강조되고 있다. 네 번째는 사교성이다. 에이전트는 상호간에 메시지를 전달할 수 있는 특성을 말한다. 이러한 특징을 만족하기 위해서 KQML과 같은 에이전트 상호간의 통신 언어에 대한 연구가 활발하게 진행되고 있다(최중민 1997).

사례기반 추론은 과거의 특정 문제에 대한 해결방안으로 사용되었던 해를 사례로 저장하며 새로운 문제에 대한 해를 추론하기 위해서 과거의 사례를 이용하는 방법이다. 사례기반 학습은 사례기반 추론과는 달리 적응성이 아닌 학습에 중점을 두고 있다. 사례기반 학습에서는 새로운 문제에 대한 해를 찾아내기 위해서 기존에 이미 저장되어 있는 사례베이스를 이용하여 해를 추출해 낼 수 있으며 사례베이스에 존재하지 않는 새로운 문제일

경우에는 새로운 사례로 사례베이스에 추가할 수 있다. 또한 이미 사례가 존재하고 있다고 해도 학습알고리즘에 따라서 새로운 사례로 수정될 수도 있다(Claire 1996).

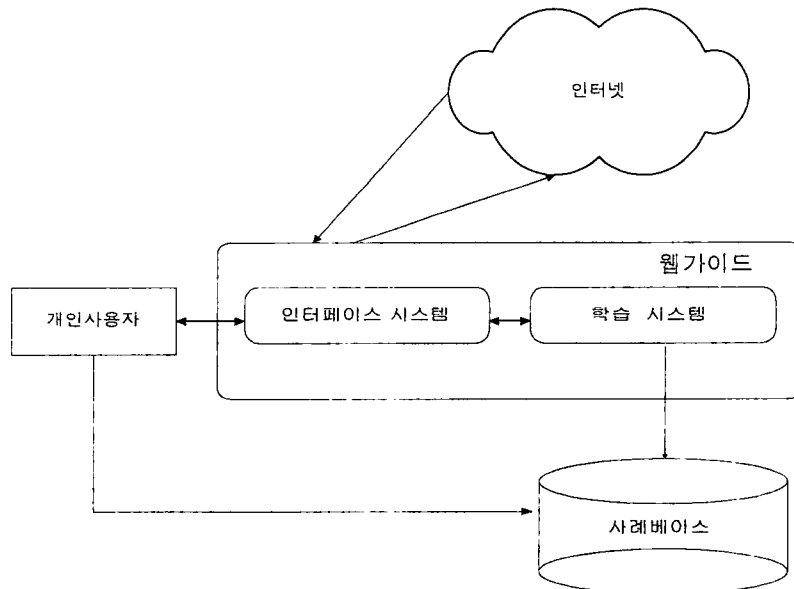
3 웹 인터페이스 에이전트 시스템

본 논문에서 제안하고 있는 웹 가이드(Web-Guide)는 인터넷에서 개인의 관심분야를 학습하여 다음 사이트를 추천해주는 개인형 웹 인터페이스 에이전트 시스템이다. 웹 가이드는 사용자가 입력한 키워드를 중심으로 한 사례기반 학습을 이용하여 개인의 사례가 자신의 시스템에

존재하는 클라이언트 기반의 시스템이다. 웹 가이드는 사용자와의 인터페이스를 담당하는 인터페이스 시스템(Interface System)과 사용자의 방문 사이트를 학습하는 학습 시스템(Learning System)으로 구성된다.

3.1 웹 에이전트 시스템

인터페이스 시스템은 사용자와 웹 가이드 사이의 모든 인터페이스 작업을 담당한다. 사용자는 인터페이스 시스템을 통해서 웹브라우징과 같은 기본적인 작업을 수행할 수 있으며 학습 시스템을 통해서 나온 추천 사이트 등을 검색할 수 있다. 또한 현재 사용자가 방문중인 사이트에 대해서 사용자 스스로가 바로 평가하여 사례베이스를 갱신할 수 있다. 학습 시스



〈그림 1〉 웹 인터페이스 에이전트 시스템

템은 인터페이스 시스템과의 통신을 통해서 사용자가 방문한 사이트들에 대해서 각각 새로운 사례들을 학습 알고리즘을 통해 학습하여 사례베이스를 갱신하는 작업을 수행한다. 웹 인터페이스 에이전트 시스템은 <그림 1>과 같다.

인터페이스 시스템에서는 먼저 사용자로부터 자신이 관심 있어 하는 키워드를 입력받는다. 사용자로 키워드를 입력받은 웹 가이드는 현재 저장되어 있는 사례베이스로부터 적정사례를 추출해 내어 임계값 이상의 사례가 사례베이스에 존재하면 사용자에게 추천한다. 사용자는 초기 추천 사이트로부터 원하는 사이트를 방문할 수 있다. 인터페이스 시스템은 사용자가 방문하고 있는 현재 페이지에 존재하고 있는 각각의 링크들을 추출, 기존에 사례베이스에 존재하는 사례들과 비교하여 임계값 이상의 링크들이 사례베이스에 존재하면 링크들을 사용자에게 추천한다. 또한 인터페이스 시스템은 현재 페이지에 있는 링크의 URL과 링크를 가리키고 있는 제목, 그리고 현재 사용자가 입력한 키워드를 추출, 학습 시스템에 전송한다.

학습 시스템은 인터페이스 시스템으로부터 전송 받은 자료가 저장되어 있는 학습 큐(Learning Queue)로부터 URL을 하나씩 추출하여 차례로 방문한다. 학습 시스템은 방문한 URL로부터 전송 받은 HTML 문서에 대해서 평가 값을 평가 함수를 통해 결정한 후 학습 알고리즘에 따라 사례에 저장한다. 이러한 작업은 학

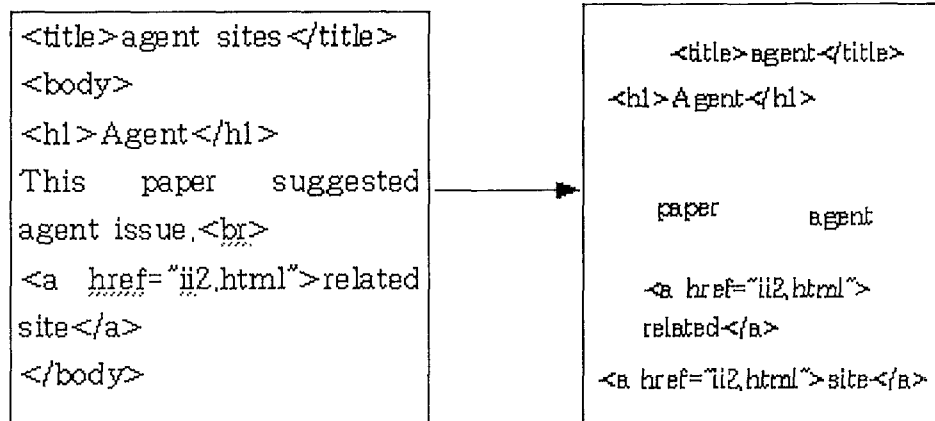
습 큐(Learning Queue)에 링크가 없어질 때까지 반복하며 수행된다. 학습 시스템에서 수행되는 작업은 수행 속도의 향상을 위해서 네트워크의 사용빈도가 떨어지는 야간작업에 주로 수행한다.

일반적으로 정보검색이나 텍스트 학습에서 하나의 문서를 표현하기 위해서는 TF·IDF-vector 표현 방식을 많이 이용한다. TF·IDF-vector 표현 방식은 하나의 문서에 대해서 단어의 순서나 구조에 관계없이 단순히 하나의 문서를 단어들의 모임(bag-of-words)으로 간주하는 방식이다(Dunja 1996). 이 방식을 이용하면 HTML과 같이 정형화가 잘 되어있는 문서나 일반 문서에서 유용하게 적용할 수 있다.

본 논문에서는 이러한 방식을 기본으로 하여 문서의 구조를 검사하는 방식을 첨가하였다. 일반적으로 HTML 문서들은 일반적인 문서와는 달리 각 단어의 특징을 표현할 수 있는 태그들을 포함하고 있다. HTML 문서에서 사용하는 태그들은 "<"로 시작하며 ">"로 종료하는 구조를 가지고 있다. 그러므로 웹 가이드는 각 단어가 포함하고 있는 태그까지 하나의 단어로 분류한다. <그림 2>는 본 논문에 이용된 HTML 문서 표현방식이다.

3.2 정보추출방법

하나의 문서에서 정보를 추출하는 방법



〈그림 2〉 본 논문에 이용된 문서 표현 방식

으로 TF·IDF 방식을 많이 사용한다. TF·IDF 방식이란 하나의 문서 d에서 단어 w에 대한 weight 값을 산출하는 방식으로 다음의 수식으로 표현할 수 있다 (Armstrong 1996).

$$TF \cdot IDF(w,d) = TF(w,d) * \log\left(\frac{1}{DF(w)}\right)$$

TF(w,d) : 문서 d에 단어 w가 나타나는 횟수
 DF(w) : 단어 w가 들어가는 문서의 총 수
 n : 전체 문서의 총수

TF·IDF 방식을 이용하면 하나의 문서 중에서 가장 weight 값이 높은 단어가 그 문서에 키워드로 채택된다.

웹 가이드는 보다 정확한 정보 수집을 위하여 사용자로부터 관심 키워드를 직접 입력받는 방식을 이용한다. 웹 가이드는 사용자로부터 입력받은 키워드를 중심으로 하여 정보를 추출한다. 웹 가이드는

사용자가 현재 방문하고 있는 홈페이지에 있는 링크들을 미리 방문하여 각 페이지에 대해서 현재 사용자의 관심분야에 대해서 정보를 추출한다. 웹 가이드는 특정 페이지에 대해서 현재 사용자가 입력한 키워드에 대한 평가를 위해서 두 가지 방법을 이용한다.

첫 번째 방법은 사용자가 입력한 관심 키워드가 페이지에 나타나는 빈도 수를 검사하는 방법이다. 특정 페이지 P에 대해서 사용자 관심 키워드 k에 대한 평가 함수 $AV_k(P)$ 는 다음과 같이 정의된다.

$$AV_k(P) = TF(P, k)$$

TF(P, k) : 특정 페이지 P에 키워드 k가 나타나는 빈도수

두 번째 방법은 첫 번째 방식에 HTML 문서의 특징을 검사하는 방법을 추가하는 방식이다. 일반적으로 HTML 문서를 제

작하는 사람들은 자신이 외부에 알리고자 하는 내용에 대해서는 여러 가지 태그를 이용하는 방식을 많이 이용한다. 웹 가이드는 사용자가 입력한 관심 키워드가 현재 페이지 중에서 특정 태그를 사용하고 있는지를 검사하여 사용한 태그에 따라서 일정한 가중치를 부여한다. 이를 이용하여 재구성한 특정페이지 P에 대해서 사용자 관심 키워드 k에 대한 평가함수 $AV_k(P)$ 는 다음과 같다.

$$AV_k(P) = \sum TV(P, k, t)$$

$TV(P, k, t)$: 특정 페이지 P에서
키워드 k가 태그 t를
사용할때의 값

$AV_k(P)$ 함수에 사용된 태그의 가중치는 홈페이지 제작 시 사용하는 전체 태그 중에서 실험 결과 홈페이지를 제작하는 사용자가 자신의 관심여부를 가장 많이 반영하는 태그에 따라 <표 1>의 가중치를 부여하였다.

<표 1> 태그에 따른 가중치

태 그	가중치	비 고
<title></title>	10	타이틀 태그
	5	글자 크기
	5	글자를 굵게 만듦
<a>	3	링크 연결

3.3 학습 알고리즘

웹 가이드에서 사용하는 학습 알고리즘은

특정 개인 사용자로부터 관심 키워드를 입력받아 하나의 사례로 저장, 평가하여 학습하는 방법이다. 웹 가이드는 웹 가이드와 사용자가 방문한 모든 홈페이지를 하나의 사례로 간주한다. 하나의 사례에는 <표 2>의 내용이 저장된다.

<표 2> 사례의 저장내용

내 용	비 고
ID	사례의 고유번호
URL	사이트 주소 저장
Agent_Value	웹가이드의 평가값 저장
User_Value	사용자의 평가값 저장

사용자는 자신이 검색하고자 하는 키워드를 입력할 수 있으며 웹 가이드는 사용자로부터 입력받은 키워드를 인덱스로 하는 리스트를 구성한다. 웹 가이드에서 사용된 사례들은 키워드에 따라 독립적으로 구성되므로 다른 키워드와의 연관성이 있는 학습은 어렵지만 사용자의 관심분야가 정확하기만 하면 보다 빠르고 효율적인 학습과정을 수행할 수 있는 장점이 있다.

웹 가이드는 인터페이스 시스템으로부터 전송 받은 자료들은 학습시스템의 Learning Queue에 저장된다. 학습시스템은 Learning Queue에 들어 있는 각각의 URL을 방문하여 각 홈페이지에 대해서 사용자가 입력한 키워드에 대해서 앞서 언급한 평가함수를 통해서 평가과정을 수행한다.

사용자가 입력한 키워드를 키워드 리스트

```

DO WHILE Get__Case() NOT empty
  Get__Case():          // 새로운 사례를 추출한다.
  Eval__Case():         // 평가함수를 이용하여 사례를 평가한다.
  IF Exist__Keywordlist then
    // 키워드리스트에 존재하면
    Check__Case():      // 새로운 사례의 존재여부 확인
    IF Exist__Case Then
      // 사례가 이미 존재하면
      Check__Value(): // 기존의 사례의 평가 값을 검사하여
      IF Now__Case__Value > Exist__Case__Value then
        // 새로운 사례의 평가 값이 크면
        Update__Case(): // 사례의 평가 값을 수정한다.
      End IF
    ELSE
      // 사례가 존재하지 않으면
      Insert__Case(): // 새로운 사례로 입력
    End IF
  ELSE
    Create__Keywordlist():
    // 새로운 키워드리스트에 등록
  End IF
LOOP
// 새로운 사례가 없을 때 까지 반복

```

〈그림 3〉 웹 가이드 학습 알고리즘

트와 비교하여 키워드 리스트에 존재하지 않을 경우에는 키워드 리스트에 추가하며 키워드를 인덱스로 하는 새로운 사례를 생성한다. 사용자가 입력한 키워드가 키워드 리스트에 존재할 경우에는 입력될 사례가 기존 사례의 존재 여부를 평가한다. 만일 새로운 사례로 판명되면 새로운 사례로 사례베이스에 추가하며 이미 존재하는 사례라면 현재 사례베이스의 사례와 새롭게 추가되어야 할 사례를 비교하여 평가 값이 우수한 사례를 수정한다.

사용자 자신이 특정 페이지에 대한 평가를 할 경우 웹 가이드는 해당 페이지를 좋은 사례로 평가하며 기존에 사용자가 평가한 사용자 평가 값의 누적된 결과에 현재 평가 값을 더한 값으로 수정된다.

이 과정을 알고리즘으로 표현하면 〈그림 3〉과 같다.

또한 특정 사용자가 평가하는 해당 페이지 p 에 대한 키워드 k 의 평가함수 $UV_k(p)$ 는 다음과 같다.

$$UV_k(p) = \sum OUV_k(p) + CUV_k(p)$$

OUV_k(p) : 키워드 k 에 대한 페이지 p 의 기존의 평가 값

CUV_k(p) : 키워드 k 에 대한 페이지 p 의 현재 평가 값

웹 가이드에서 사례는 웹 가이드가 방문한 URL에 대한 평가 값과 사용자가 현재 URL에 대한 평가 값을 복합적으로 저장한다.

결과적으로 웹 가이드가 평가하는 특정 페이지 p에 대한 키워드 k의 평가 값 V_k(p) 는 다음과 같다.

$$V_k(p) = AV_k(p) + UV_k(p)$$

AV_k(p) : 페이지 p 에 키워드 k 에 대한 웹가이드의 평가값

UV_k(p) : 페이지 p 에 키워드 k 에 대한 사용자의 평가값

3.4 검색 방식

일반적으로 웹에서의 검색 방식은 기존의 인공지능에서 연구해온 검색 방법과 유사한 점이 많이 존재한다. 웹에서 하나의 페이지를 Node로 간주하고 다른 사이트를 가리키고 있는 링크들을 Edge로 간주하는 하나의 그래프로 표현할 수 있다.

본 논문에서는 Best-First 검색방식의 한 방식인 A* 알고리즘을 이용하였다. A* 알고리즘은 중복되는 경로의 탐색을

피하기 위해서 그래프를 탐색하는 효과적인 방식이다. A* 알고리즘에서 새로운 노드를 선택하기 위한 함수 f(n)로 정의할 수 있으며 본 논문에서는 앞으로 진행되어야 할 링크의 평가 값만을 필요로 하므로 g(n)은 0으로 할당되며 h(n)는 앞에서 언급한 평가함수에 의해서 할당한다.

$$f(n) = g(n) + h(n)$$

g(n) : 출발 상태부터 현재 노드까지 오는데 소요된 평가값

h(n) : 현재 노드로부터 목표노드 가는데 필요한 평가값

3.5 구현

웹 가이드는 윈도우즈 98 환경에서 인터넷 사용이 가능한 IBM-PC에서 비주얼 베이직 6.0으로 구현하였다. 웹 가이드는 인터페이스 시스템과 학습 시스템의 두 가지 부 시스템으로 구현하였다. 실제 사용자는 인터페이스 시스템만을 이용하게 된다. 웹 가이드의 사용방식은 다음과 같다.

3.5.1 인터페이스 시스템

인터페이스 시스템은 기존의 웹 브라우저와 동일한 환경을 제공하도록 구현하였으며 인터페이스 시스템은 사용자에게 현재 링크중 사용자의 관심 키워드에 대한 평가 값이 높은 순서대로 추천해 준다. 인터페이스 시스템의 운영 방식은 다음과 같다.

(1) 사용자는 <그림 4>에서 먼저 자신이 검색하고자 하는 키워드를 입력한다.

인터페이스 시스템은 사례베이스에서 사용자가 입력한 키워드의 사례들을 추출하여 평가 값이 높은 순서대로 사용자에게 <그림 5>의 초기 사이트를 추천한다.

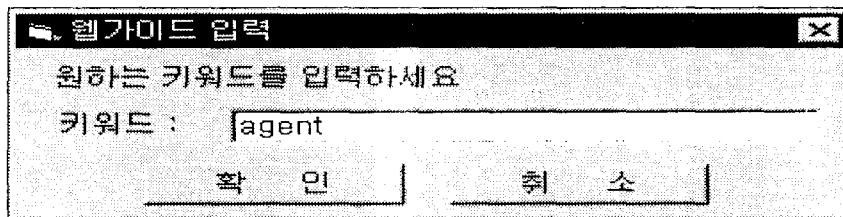
(2) 사용자는 웹 가이드가 추천하는 사이트 중에서 자신이 원하는 사이트로 이동할 수 있다.

(3) <그림 6> 는 웹 가이드 인터페이스 시스템의 주 화면이다. 웹 가이드는 현재 보여지고 있는 페이지의 모든 링크에 대해서 사례베이스와 비교하여 임계값이 높

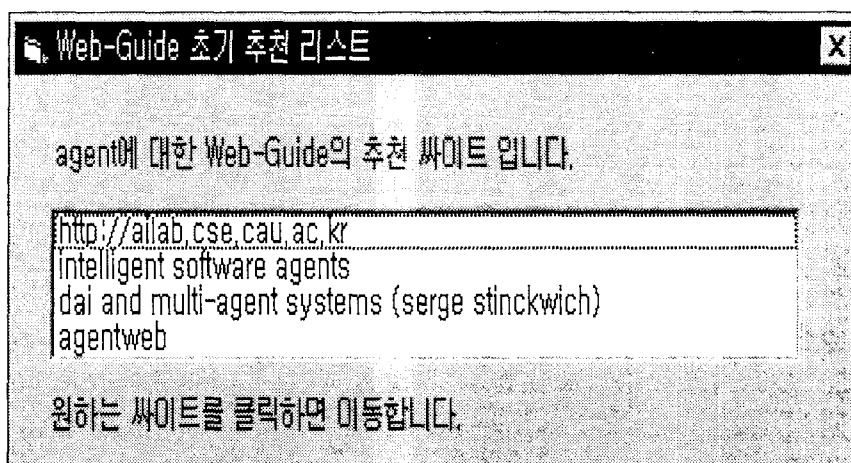
은 순서대로 특정 링크를 화면 윗 부분에 추천 사이트로 표시한다. 그리고 현재 페이지의 링크를 학습시스템으로 전송한다. 사용자는 자신이 원하는 링크를 클릭할 수 있으며 운영방식은 기존의 웹 브라우저와 동일하다. 또한 사용자는 인터페이스 시스템을 통해서 현재 페이지에 대해서 직접적으로 평가를 할 수 있다.

3.5.2 학습 시스템

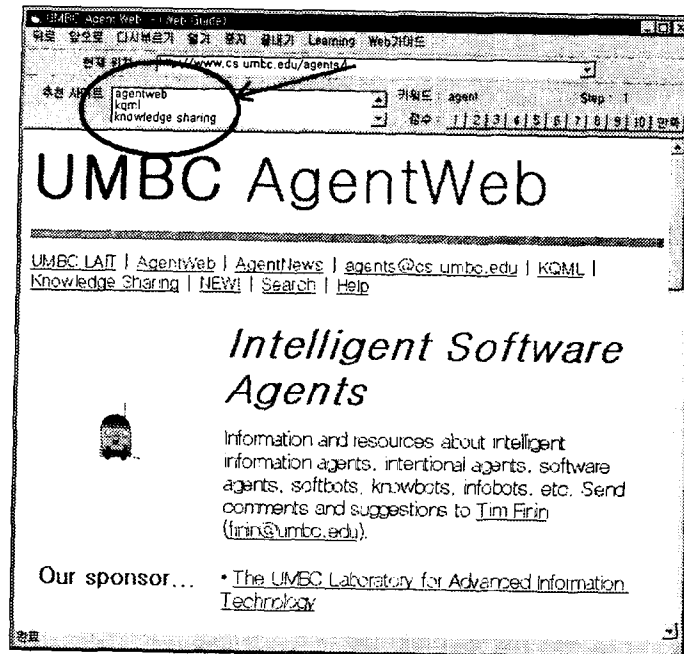
학습 시스템은 <그림 7>와 같으며 하나는 학습 시스템을 통해서 현재 학습 큐(Learning



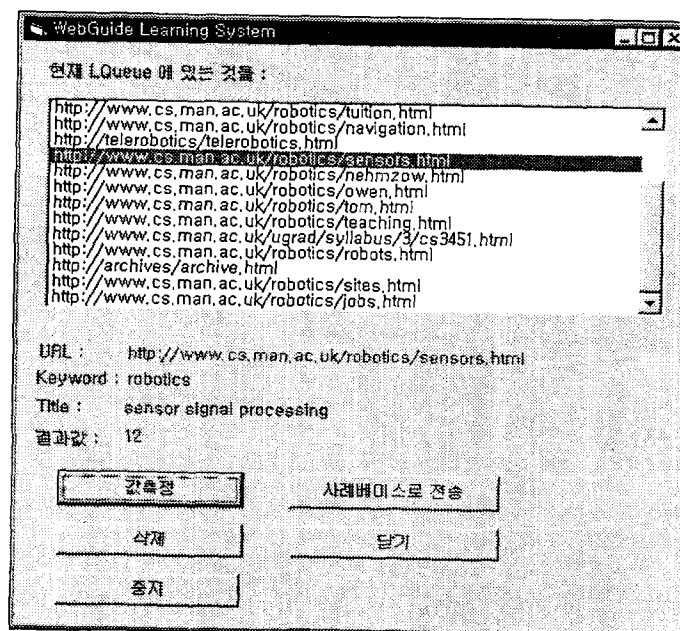
<그림 4> 키워드 입력화면



<그림 5> 웹 가이드 초기 추천 사이트 리스트 화면



〈그림 6〉 웹 가이드 인터페이스 시스템 화면



〈그림 7〉 웹 가이드 학습 시스템 화면

Queue)에 저장되어 있는 내용을 확인할 수 있으며 실시간으로 특정 페이지에 대한 평가 값을 확인 할 수도 있다. 또한 존재하지 않는 사이트는 삭제할 수 있다.

학습 시스템에는 현재 학습 큐(Learning Queue)에 저장되어 있는 사이트들을 차례로 방문하며 평가함수를 통해 사례로 저장하는 작업을 수행할 수 있다.

4 실험 및 평가

4장에서는 웹 가이드를 이용한 실험환경 및 실험 방법을 설명하며 실험을 통해 나타난 결과를 바탕으로 웹 가이드의 성능을 평가한다.

4.1 실험 환경

본 논문은 인터넷이 연결되어 있는 컴퓨터를 이용하는 특정 개인을 대상으로 실험하였다. 실험에 참가한 인원은 컴퓨터에 대한 일반적인 지식을 지닌 일반사용자와 인공지능에 대해서 적정수준 이상의 지식을 가진 전문사용자로 구분하였다.

본 논문에서 실시한 실험은 인공지능 관련 사이트만을 대상으로 하였으며 <표 3>의 키워드와 <표 4>의 사이트를 사용하였다. 또한 웹에서 이용하고 있는 멀티미디어 데이터를 배제하고 일반 HTML 문서만을 대상으로 실험하였다.

실험에 이용된 웹 가이드는 <표 3>의 키워드에 대해서 10,000 개의 사이트를

기반 사례로 이용한다.

<표 3> 실험에 이용된 키워드 리스트

Agent	Fuzzy
Genetic	Expert
Cognitive	Alife
Autonomous	Intelligent
Search	Learning
Neural	Robotics
Agents	Mining

4.2 평 가

실험에 참가한 사용자들은 각각 <표 3>에서 제시한 키워드에 대해서 <표 4>에 있는 사이트를 시작으로 자신의 관심정도에 따라 원하는 사이트를 방문하여 방문하는 페이지에 대한 정확도를 평가하는 방식이다. 사용자가 한번의 링크를 클릭해서 다음 페이지로 이동하는 것을 Step으로 정의하였으며 이전 페이지로 복귀하는 Step 수는 인정하지 않았다. 또한 하나의 페이지에 대해서 웹 가이드가 추천해주는 링크의 수는 최대 5개로 한정하였다. 본 실험에 사용된 평가함수는 태그의 가중치를 평가하는 $AV^k(p)$ 이다. 사용자는 정해진 Step 수에 따라 사이트를 방문하며 방문한 사이트에 대한 정확도를 부여하는 방식이다.

첫 번째 실험방법은 인공지능에 대한 특별한 지식이 없는 일반사용자를 대상으로 실험하였다. 사용자는 주어진 키워드

〈표 4〉 실험에 사용된 사이트 리스트

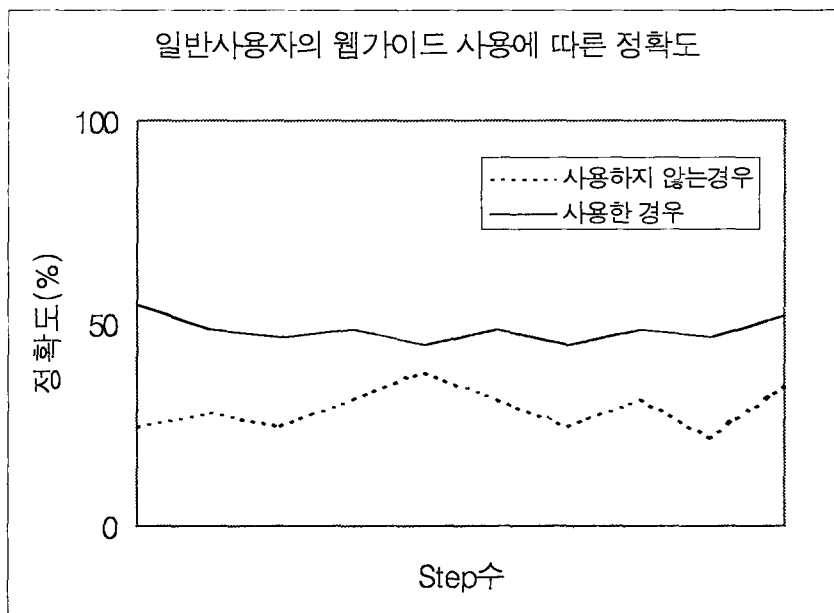
http://www.cs.washington.edu/research/projects/ai/www/
http://www.ai.mit.edu/
http://www.cs.mu.oz.au/~leon/agentlinks.html
http://www.isi.edu/soar/tambe/agent.html
http://www.cs.umbc.edu/agents/
http://www.comp.ae.keio.ac.jp/lab/okoma/cho/alink.html
http://www.sics.se/isl/abc/survey.html
http://www.krl.caltech.edu/avida/
http://www.soton.ac.uk/~adc96r/
http://www.dai.ed.ac.uk/
http://mojave.eecs.uic.edu/
http://alma.uc.pt/
http://ece.clemson.edu/iaal/
http://www.eng2.uconn.edu/BRC/ARTIFICIAL/ai_lab.html
http://www.libs.uga.edu/science/artintel.html
http://www.cs.msstate.edu/artificial_intelligence/
http://www.wins.uva.nl/research/neuro/
http://www.comlab.ox.ac.uk/archive/comp/ai.html
http://www-mice.cs.ucl.ac.uk/misc/ai/
http://www-cs.stanford.edu/profile/ai.html
http://krusty.eecs.umich.edu/
http://www.cs.man.ac.uk/ai/
http://www.cs.wisc.edu/~shavlik/uwai.html
http://dcpul.cs.york.ac.uk:6666/~isg/
http://nathan.gmd.de/
http://www.cs.utexas.edu/users/ai-lab/
http://www.eeb.ele.tue.nl/neural/neural.html
http://www.mbfys.kun.nl/SNN/homepageSNN.html

에 대한 일반적인 정의를 지정된 Step 동안 검색하며 이때 웹 가이드를 사용할 경우와 그렇지 않을 경우에 따른 정확도를 비교하였다. 정확도의 기준은 사용자 자신이 해당 키워드에 대한 내용의 정확성을 0~5 사이의 값으로 평가한다.

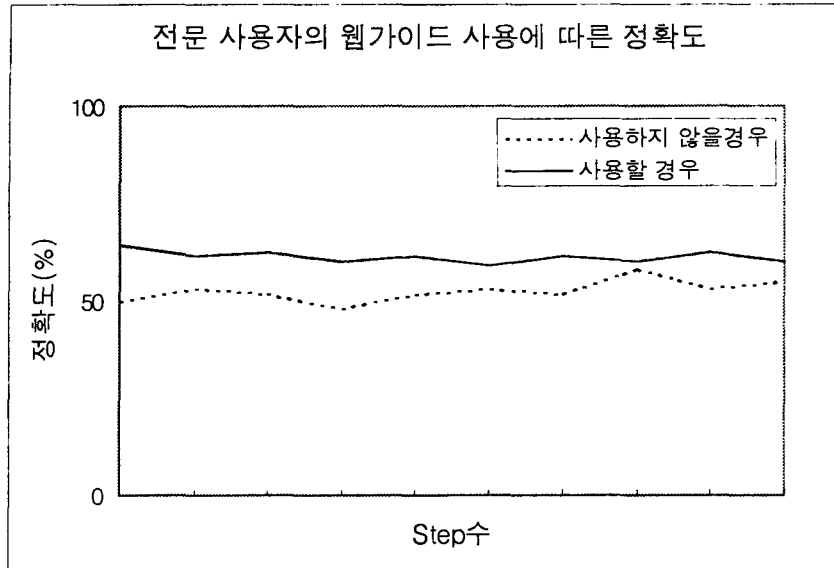
그래프는 실험에 참가한 사용자들의 평균 정확도를 표시하고 있다. 웹 가이드를 사용하지 않은 일반사용자는 정확도가 비교적 낮았다. 웹 가이드를 사용한 경우에는 그렇지 않은 경우보다 전체적으로 정확도가 높았으며 Step 수의 변화에 따른 정확도의 변화 폭이 감소하였다. 이는 사례베이스를 가지고 있는 웹 가이드가 특정 사이트를 추천하는 경우 전문분야에 특별한 전문지식이 없는 일반 사용자들도 웹

가이드가 자신이 원하는 사이트를 검색하는 데 상당한 도움을 주었음을 입증한다.

또한 첫 번째 Step에서는 정확도의 편차가 심하게 나타나는 결과를 보였다. 이는 사용자가 원하고자 하는 정보를 얻는데 첫 번째 사이트가 중요한 역할을 하는 것을 보여준다. 두 번째 실험방법은 인공지능에 대한 기반 지식이 있는 전문사용자를 대상으로 실험하였다. 사용자는 주어진 키워드에 대한 일반적인 정의를 지정된 Step 동안 검색하며 이때 웹 가이드를 사용할 경우와 그렇지 않을 경우에 따른 정확도를 비교하였다. 사용된 평가함수와 정확도의 기준은 첫 번째 실험과 동



〈그림 8〉 일반사용자의 웹 가이드 사용여부와 정확도 관계



〈그림 9〉 전문 사용자의 웹 가이드 사용여부와 정확도 관계

일하다.

전문 사용자의 경우에도 웹 가이드를 사용하였을 때의 정확도가 웹 가이드를 사용하지 않았을 때보다 높은 결과를 나타내었다. 또한 정확도의 변화폭도 웹 가이드를 사용하였을 때가 웹 가이드를 사용하지 않았을 때보다 작은 수치로 나타났다.

하지만 첫 번째 실험에 비해서 웹 가이드의 사용여부가 사용자의 정확도평가에는

적게 영향을 미쳤다. 이는 전문사용자의 기반지식과 웹 가이드의 사례베이스에 존재하는 지식의 공유부분이 많이 존재하기 때문이다.

각 실험에 대해서 사용자의 평균정확도는 〈표 5〉과 같다.

평균 정확도는 실험에 참가한 각 개인의 정확도를 평균값으로 산정 한 것으로 랜덤의 경우 평균 18.6%의 정확도를 나

〈표 5〉 각 사용자의 평균 정확도

종 류	평균 정확도(%)
랜덤	18.6
일반 사용자가 웹가이드를 이용하지 않는 경우	29.3
일반 사용자가 웹가이드를 이용하는 경우	45.6
전문 사용자가 웹가이드를 이용하지 않는 경우	52.6
전문 사용자가 웹가이드를 이용하는 경우	61.3

타내었다. 일반 사용자가 웹 가이드를 이용했을 경우에는 그렇지 않았을 경우보다 약 11%정도의 정확도 향상이 있었으며 전문 사용자의 경우에는 약 8.6% 정도의 정확도 향상이 있었다. 하지만 일반 사용자가 웹 가이드의 추천을 이용하였을 경우보다 전문사용자가 자신의 지식을 바탕으로 검색하였을 경우에는 전문사용자의 정확도가 우수한 것으로 평가되었다. 이는 웹 가이드가 학습한 기반 지식이 전문 사용자의 기반 지식에는 이르지 못한 것을 보여준다.

5 결 론

인터넷에 존재하는 많은 정보 중에서 사용자 자신이 원하는 정보를 찾고자 하는 연구가 활발히 진행되었다. 하지만 검색 엔진과 같은 기존의 방식에서는 검색된 결과가 필요이상으로 많이 나타나고 불특정 다수를 대상으로 하기 때문에 자신이 원하는 결과를 찾기가 어려웠다. 본 논문에서는 이러한 문제를 해결하기 위해서 특정 개인 사용자를 대상으로 하는 인터넷 웹 인터페이스 에이전트에 대해서 연구하였다. 본 논문에서 제안하고 있는 웹 가이드는 특정 개인의 관심 분야를 지속적으로 관찰, 학습하는 에이전트로 개인의 관심분야만을 집중적으로 학습한다. 웹 가이드는 인터페이스 시스템과 학습 시스템의 두 개의 부 시스템으로 이루어져 있다. 하나의 HTML 문서를 표현하기 위해서 하나의

문서에 나타난 단어를 Vector 방식으로 표현하는 TF·IDF-Vector 방식에 태그를 추가하여 검사하는 방식을 이용하였다. 문서의 가중치를 평가하는 정보추출 방식으로는 사용자가 입력한 키워드가 문서에 나타나는 빈도 수를 평가하는 방식과 키워드에 이용된 태그에 가중치를 부여하여 평가하는 방식을 이용하였다. 학습 방법으로는 평가함수를 통해 평가된 사례를 기반으로 한 사례기반 학습을 이용하였다. 실험결과 일반사용자와 전문사용자 모두 자신이 원하는 정보를 웹 가이드를 이용하였을 경우가 그렇지 않았을 경우보다 빠르게 얻을 수 있었다. 또한 키워드에 가중치를 두어 평가한 평가함수를 이용한 웹 가이드가 그렇지 않은 경우보다 좋은 결과를 얻었다. 결과적으로 웹 가이드를 이용하면 사용자는 자신만의 관심 분야에 대해서 보다 빠른 시간 내에 원하는 결과를 얻을 수 있게 된다.

향후 연구로는 사례베이스간의 관련성을 추가한 사례베이스 설계방법과 적절한 사례베이스의 유지방법에 대한 연구가 필요하며 복합 키워드의 처리에 대한 연구가 요구된다.

참 고 문 헌

- 최중민, 1997. 에이전트의 개요와 연구방향. 『정보과학회지』, 15(3): 8~10.

- Armstrong, R., D. Freitag, T. Joachims, and T. Mitchell. 1996. "Web-Watcher: a learning apprentice for the world wide web." *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous*. Distributed Environments, Stanford.
- Balabanovic, Marko, Yoav Shoham. 1996. *Learning Information Retrieval Agents: Experiments with Automated Web Browsing*. Stanford University.
- Cardie, Claire. 1996. *A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis*. Department of Computer Science University of Massachusetts Amherst, MA 01003.
- Mladenic, Dunja. 1996. *Personal Web-Watcher: Design and Implementation*. Carnegie Mellon University.