

# 사례기반 학습을 이용한 개인형 웹 에이전트 시스템

김 영 권<sup>†</sup> · 이 재 필<sup>†</sup> · 이 말 레<sup>†</sup> · 김 경 만<sup>††</sup> · 김 기 태<sup>†††</sup>

## 요 약

최근 인터넷의 정보가 폭발적으로 증가하고 있다. 사용자들은 인터넷상의 수 많은 정보중에서 자신이 원하는 정보를 찾기 원하지만 유용한 정보를 찾기란 쉬운 일이 아니다.

본 논문은 이러한 문제를 효율적으로 해결하기 위하여 사례기반 학습을 이용한 웹가이드인 개인형 웹 에이전트 시스템을 제안한다. 웹가이드는 인터페이스 시스템과 학습 시스템의 두 개의 부시스템으로 이루어져 있으며, 다른 웹 브라우저와 같은 작업을 수행하며 사용자를 시스템에 연결시켜 준다. 또한 인터페이스 시스템은 현재 페이지에 있는 데이터를 학습 큐에 전달한다. 학습 시스템은 학습 큐에 있는 각 페이지를 평가함수를 이용하여 방문하며, 가중치는 HTML 문서의 문자로 사용된 태그를 분석하여 결정된다.

실험 결과 일반 사용자와 전문 사용자 모두 자신이 원하는 정보를 웹가이드를 이용하였을 경우가 그렇지 않았을 경우보다 빠르게 얻을 수 있었다.

## A Personal Web-Agent System Using Case-Based Learning

Young-Kwon Kim<sup>†</sup> · Jae-Pil Lee<sup>†</sup> · Mal-Rey Lee<sup>†</sup> · Kyung-Man Kim<sup>††</sup> · Ki-Tae Kim<sup>†††</sup>

## ABSTRACT

Recently, massive amount of information is provided for the internet users. So users want to search information on Internet, but it is difficult to search information what you want.

In this paper, we propose a personal Web-agent system using Case-based learning, Web-Guide. Web-Guide consists of two sub-system, interface-system and learning-system. Interface-system operates other web-browser nearly the same and connects user to system. And interface-system transfer datas of current page to learning-queue. Learning-system visit and evaluate the value of each page in learning-queue using evaluation-function that gave weight values occupied by analyzing tag used the character of HTML document.

After all users who are known about artificial intelligence well and not made experiments by using Web-Guide, they reached their desired sites faster than before.

### 1. 서 론

최근 인터넷의 폭발적인 사용 증가로 인해 사용자는 하루에도 셀수 없을 정도로 많은 양의 정보를 접하게

된다. 정보가 늘어감에 따라 인터넷을 이용하는 사용자들은 이러한 수많은 정보중에서 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 사이트에 대한 정보를 저장하고 있는 검색엔진을 이용하는 경우가 대부분이다. 하지만 자료의 양이 방대해짐에 따라 검색엔진을 통한 결

<sup>†</sup> 준 회 원 : 중앙대학교 컴퓨터공학과 대학원

<sup>††</sup> 준 회 원 : 온네트 선임연구원

<sup>†††</sup> 정 회 원 : 중앙대학교 컴퓨터공학과 교수

논문접수 : 1997년 2월 5일, 심사완료 : 1998년 4월 27일

과물도 사용자들에게는 불필요한 정보나 필요이상의 결과를 보여주는 경우가 빈번하게 발생하여 사용자에게 실제로 필요한 정보가 아닐 경우가 많이 발생한다[5][9].

따라서 본 논문에서는 이러한 문제를 해결할 수 있는 개인형 웹 인터페이스 에이전트 시스템인 웹가이드(Web-Guide)를 제안한다. 웹가이드는 사용자의 행동과 에이전트의 방문을 키워드를 중심으로 각각의 사례로 저장하는 사례기반 학습 방법을 이용하여 특정 개인 사용자가 웹상에서 검색하고자 하는 자료를 입력받은 후부터 사용자의 방문 행동을 학습하여 보다 빠른 시간 내에 원하고자 하는 자료를 검색할 수 있도록 도와주는 에이전트 시스템이다[6][7][8].

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 관련 연구로써 기존의 웹 에이전트에 대해서 간략히 설명한다. 3장에서는 본 논문에서 제안한 웹가이드 시스템의 전체 구성 및 문서 표현방법에 대하여 설명한다. 4장에서는 웹가이드 시스템의 실험환경 및 실험을 통해 나온 결과를 평가한다. 5장에서는 4장에서 나온 결과를 기반으로 본 논문에 대한 결론을 맺으며 향후 연구과제에 대해 소개한다.

## 2. 관련연구

### 2.1 WebWatcher

WebWatcher는 사용자의 관심키워드를 입력받은 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 WebWatcher는 이를 기준으로 다음 사용자에게 추천해주는 방식을 이용한다. WebWatcher에서는 문서를 표현하는 방식으로 하나의 문서를 단어단위로 분류하여 Boolean 값을 갖는 방법을 이용하였고 평가 함수로는 TFIDF를 이용하였다[1]. 그러나 이 연구는 대상자를 불특정 다수로 하여 하나의 키워드에 대해서 사용자 각자가 서로 다른 관점을 가질 수 있으므로 사용자는 추천받은 링크로부터 자신이 원하는 정보를 얻을 수 없는 경우가 발생할 수 있다.

### 2.2 Personal WebWatcher

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 방식은 Web-Watcher와 동일하며 평가함수는 베이지언 분류(Bayesian-classifier) 방식을 이용하였다[2]. 그러나 이 연구는 사용자가 자신의 관심여부를 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.

### 2.3 LIRA

Marko Balabanovic이 제안한 LIRA는 특정 정보를 검색하는 것이 아닌 일반적으로 써핑하는 것에 중점을 두었다. 사용자의 일반적인 웹서핑을 기반으로 하여 휴리스틱한 방법으로 특정 페이지들을 추천해 준 후 사용자로부터 결과값을 평가하여 새로운 페이지를 추천해주는 방식을 이용하였다. 또한 LIRA는 문서를 벡터(Vector) 방식으로 표현하였다[3]. 그러나 이 연구 역시 사용자의 관심여부를 명확히 파악할 수 없기 때문에 사용자에게 비교적 정확한 사이트를 추천하기 위해서는 학습기간이 오래걸린다는 문제점이 있다.

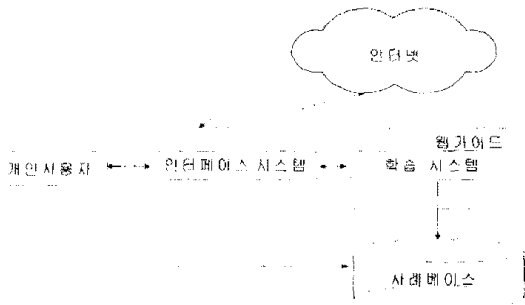
본 논문은 이러한 문제점을 해결하기 위하여 사용자와 에이전트의 사례를 복합적으로 평가하는 사례기반학습을 이용하였으며, 학습기간의 단축 및 보다 정확한 사이트의 추천을 위하여 사용자의 대상을 특정개인으로 한정하여 키워드의 관심여부를 보다 명확히 하였다. 또한 하나의 문서를 표현하기 위해서 WebWatcher와 동일한 방법을 이용하였으나 문서의 평가값을 순차적으로 표현할 수 있는 방식을 채택하였으며 키워드에 이용된 태그를 복합적으로 평가하는 평가함수를 채택하였다.

## 3. 웹가이드

### 3.1 구조

웹가이드의 구조는 (그림 1)과 같다. 인터페이스 시스템은 사용자와 웹가이드 사이의 모든 인터페이스 작업을 담당한다. 사용자는 인터페이스 시스템을 통해서

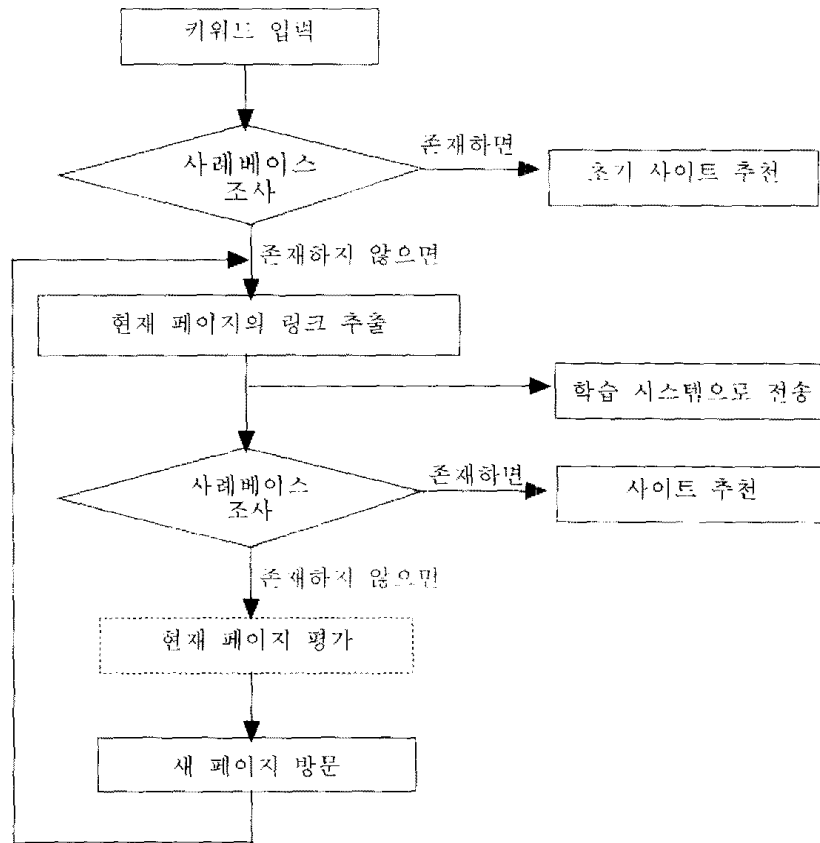
웹브라우저와 같은 기본적인 작업을 수행할 수 있으며 학습 시스템을 통해서 나온 추천 사이트들을 검색할 수 있다. 또한 현재 사용자가 방문중인 사이트에 대해서 사용자 스스로가 직접 평가하여 사례베이스를 갱신할 수 있다. 학습 시스템은 인터페이스 시스템과의 통신을 통해서 사용자가 방문한 사이트들에 대해서 각각 새로운 사례들을 학습 알고리즘을 통해 학습하여 사례베이스를 갱신하는 작업을 수행한다.



(그림 1) 웹가이드의 구조  
(Fig. 1) Structure of Web-Guide

### 3.1.1 인터페이스 시스템

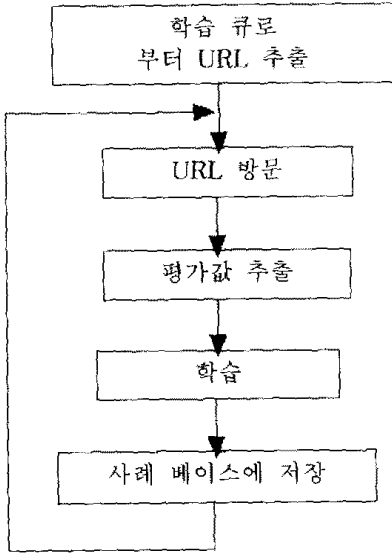
인터페이스 시스템의 수행과정은 (그림 2)와 인터페이스 시스템에서는 먼저 사용자로부터 자신이 관심있는 키워드를 입력받는다. 사용자로 키워드를 입력받은 웹가이드는 현재 저장되어 있는 사례베이스로부터 적성 사례를 추출해 내어 임계값 이상의 사례가 사례베이스에 존재하면 사용자에게 추천한다. 사용자는 초기 추천 사이트로부터 원하는 사이트를 방문할 수 있다. 인터페이스 시스템은 사용자가 방문하고 있는 현재 페이지에 존재하고 있는 각각의 링크들을 추출, 기존에 사례베이스에 존재하는 사례들과 비교하여 임계값 이상의 링크들이 사례베이스에 존재하면 링크들을 사용자에게 추천한다. 또한 인터페이스 시스템은 현재 페이지에 있는 링크의 URL과 링크를 가리키고 있는 제목, 그리고 현재 사용자가 입력한 키워드를 추출, 학습 시스템에 전송한다.



(그림 2) 인터페이스 시스템  
(Fig. 2) Interface System

3.1.2 학습 시스템

학습 시스템의 수행과정은 (그림 3)과 같다.

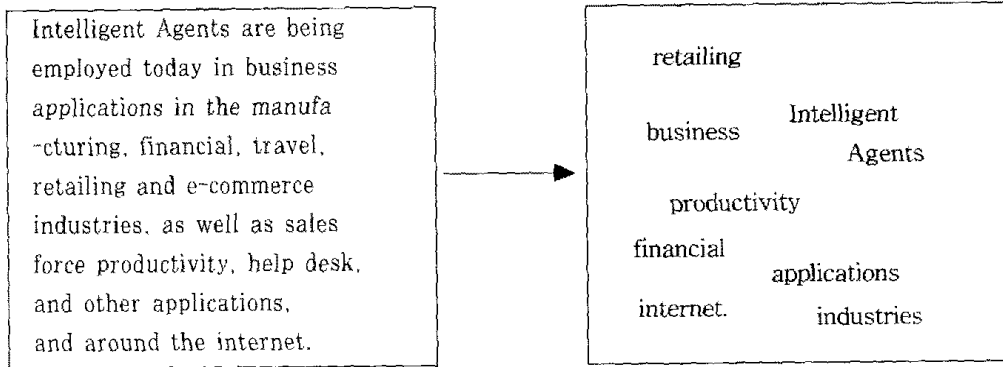


(그림 3) 학습 시스템  
(Fig. 3) Learning System

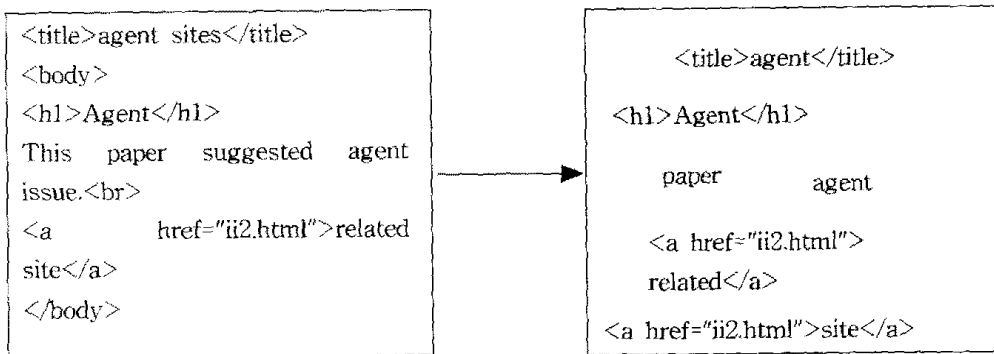
학습 시스템은 인터페이스 시스템으로부터 전송받은 자료가 저장되어 있는 학습큐(Learning Queue)로부터 URL을 하나씩 추출하여 차례로 방문한다. 학습 시스템은 방문한 URL로부터 전송받은 HTML 문서에 대해서 평가값을 평가 함수를 통해 결정한 후 학습 알고리즘에 따라 사례에 저장한다. 이러한 작업은 학습큐에 링크가 없어질 때 까지 반복하며 수행된다.

3.2 문서 표현방법

일반적으로 정보검색이나 텍스트 학습에서 하나의 문서를 표현하기 위해서는 TFIDF-vector 표현 방법을 많이 이용한다. TFIDF-vector 표현 방법은 하나의 문서에 대해서 단어의 순서나 구조에 관계없이 단순히 하나의 문서를 단어들의 모임으로 간주하는 방법이다 [2][4]. 이 방법을 이용하면 HTML 과 같이 정형화가 잘 되어있는 문서나 일반 문서에서 유용하게 적용할 수 있다.



(그림 4) TFIDF-Vector 표현 방법  
(Fig. 4) Representation Method of TFIDF-Vector



(그림 5) 문서 표현 방법  
(Fig. 5) Representation Method of Document

본 논문에서는 이러한 방법을 기본으로 하여 문서의 구조를 검사하는 방법을 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 웹가이드는 각 단어가 포함하고 있는 태그까지 하나의 단어로 분류한다.

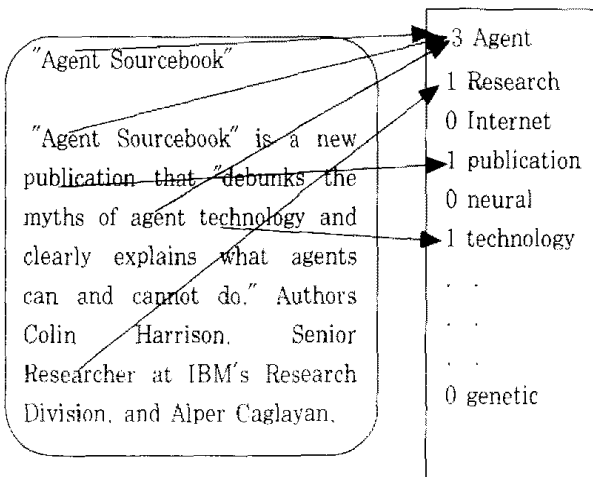
3.3 정보 추출방법

하나의 문서에서 정보를 추출하는 방법으로 TFIDF 방법을 많이 사용한다. TFIDF 방법이란 하나의 문서 d에서 단어 w에 대한 weight 값을 산출하는 방식으로 다음의 수식으로 표현할 수 있다[1].

$$TFIDF(w,d) = TF(w,d) * \log\left(\frac{n}{DF(w)}\right)$$

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

TFIDF 방법을 이용하면 하나의 문서 중에서 가장 weight 값이 높은 단어가 그 문서에 키워드로 채택된다.



(그림 6) TFIDF 표현 방법  
 (Fig. 6) Representation Method of TFIDF

웹가이드는 보다 정확한 정보 수집을 위하여 사용자로부터 관심 키워드를 직접 입력받는 방법을 이용한다. 웹가이드는 사용자로부터 입력받은 키워드를 중심으로 하여 정보를 추출한다. 웹가이드는 사용자가 현재 방문

하고 있는 홈페이지에 있는 링크들을 미리 방문하여 각 페이지에 대해서 현재 사용자의 관심분야에 대해서 정보를 추출한다. 웹가이드는 특정 페이지에 대해서 현재 사용자가 입력한 키워드에 대한 평가를 위해서 두가지 방법을 이용한다.

첫 번째 방법은 사용자가 입력한 관심 키워드가 페이지에 나타나는 빈도수를 검사하는 방법이다. 특정 페이지 P에 대해서 사용자 관심 키워드 k에 대한 평가함수  $AV_k(P)$ 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 방법에 HTML 문서의 특징을 검사하는 방법을 추가하는 방법이다. 일반적으로 HTML 문서를 제작하는 사람들은 자신이 외부에 알리고자 하는 내용에 대해서는 여러 가지 태그를 이용하는 방식을 많이 이용한다. 웹가이드는 사용자가 입력한 관심 키워드가 현재 페이지 중에서 특정 태그를 사용하고 있는 지를 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 제작시 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  
 (Table 1) Weight value of tag

태그	가중치	비고
<title></title>	10	타이틀 태그
<h1></h1>	5	글자 크기
<b></b>	5	글자를 굵게 만들
<a></a>	3	링크 연결

### 3.4 사례기반 학습 알고리즘

웹가이드에서 사용하는 학습 알고리즘은 특정 개인 사용자로부터 관심 키워드를 입력받아 하나의 사례로 저장, 평가하여 학습하는 방법이다. 웹가이드는 웹가이드와 사용자가 방문한 모든 홈페이지를 하나의 사례로 간주한다. 하나의 사례에는 <표 2>의 내용이 저장된다.

<표 2> 사례의 표현  
<Table 2> Representation of case

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 웹가이드는 사용자로부터 입력받은 키워드를 인덱스로 하는 리스트를 구성한다. 웹가이드에서 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.

웹가이드는 인터페이스 시스템으로부터 전송받은 자료들을 학습시스템의 학습 큐에 저장한다. 학습시스템은 학습 큐에 들어 있는 각각의 URL을 방문하여 각 홈페이지에 대해서 사용자가 입력한 키워드에 대해서 앞서 언급한 평가함수를 통해서 평가과정을 수행한다. 평가과정을 수행한 하나의 사례를 사례베이스에 저장하는 과정은 다음과 같다. 사용자가 입력한 키워드를 키워드 리스트와 비교하여 키워드 리스트에 존재하지 않을 경우에는 키워드 리스트에 추가하며 키워드를 인덱스로 하는 새로운 사례를 생성한다. 사용자가 입력한 키워드가 키워드 리스트에 존재할 경우에는 입력된 사례가 기존 사례의 존재 여부를 평가한다. 만약 새로운 사례로 판명되면 새로운 사례로 사례베이스에 추가하며 이미 존재하는 사례라면 현재 사례베이스의 사례와 새롭게 추가되어야 할 사례를 비교하여 평가값이 우수한 사례를 수정한다. (그림 7)은 웹가이드의 학습 알고리즘이다.

```

DO WHILE Get_Case() NOT empty
  Get_Case():           // 새로운 사례를 추출한다.
  Eval_Case():         // 평가함수를 이용하여 사례를 평가
                      // 한다.
  IF Exist_Keywordlist then
                      // 키워드리스트에 존재하면
    Check_Case():     // 새로운 사례의 존재여부 확인
    IF Exist_Case Then
                      // 사례가 이미 존재하면
      Check_Value(): // 기존의 사례의 평가값을 검사하여
                      // 새로운 사례의 평가값이 크면
      Update_Case(): // 사례의 평가값을 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 7) 웹가이드 학습 알고리즘  
(Fig. 7) Learning algorithm of Web-Guide

또한 웹가이드는 특정 개인을 대상으로 하는 시스템이므로 사용자의 평가가 학습에 중요한 영향을 미치게 된다. 사용자는 자신이 현재 방문하고 있는 임의의 페이지에 대해서 평가를 할 수 있으며 웹가이드는 사용자의 평가에 따라 현재 페이지에 대한 사례를 수정한다. 현재 페이지에 대한 사용자의 평가를 학습하는 알고리즘은 (그림 8)과 같다.

```

IF Check_Case() then // 현재 사례베이스에 존재하면
  Case_Update():     // 사례를 수정
ELSE                // 사례베이스에 존재하지 않으면
  Create_Case():
                      // 입력받은 키워드를 값으로 하는
                      // 사례 생성
  Insert_Case():     // 새로운 사례로 입력
END IF
    
```

(그림 8) 사용자에게 의한 사례학습 알고리즘  
(Fig. 8) Case learning algorithm by user

사용자 자신이 특정 페이지에 대한 평가를 할 경우 웹가이드는 해당 페이지를 좋은 사례로 평가하며 기존에 사용자가 평가한 사용자 평가값의 누적된 결과에 현재 평가값을 더한 값으로 수정된다.

특정 사용자가 평가하는 해당 페이지 p에 대한 키워드 k의 평가함수  $UV_k(p)$ 는 다음과 같다.

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

$OUV_k(p)$  : 키워드 k에 대한 페이지 p의 기존의 평가값  
 $CUV_k(p)$  : 키워드 k에 대한 페이지 p의 현재 평가값

웹가이드에서 사례는 웹가이드가 방문한 URL에 대한 평가값과 사용자가 현재 URL에 대한 평가값을 복합적으로 저장한다. 결과적으로 웹가이드가 평가하는 특정 페이지 p에 대한 키워드 k의 평가값  $V_k(p)$ 는 다음과 같다.

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

$AV_k(p)$  : 페이지 p에 키워드 k에 대한 웹가이드의 평가값  
 $UV_k(p)$  : 페이지 p에 키워드 k에 대한 사용자의 평가값

### 4. 실험 및 평가

#### 4.1 실험 환경

본 논문은 인터넷이 연결되어 있는 컴퓨터를 이용하는 특정 개인을 대상으로 실험하였다. 실험에 참가한 인원은 컴퓨터에 대한 일반적인 지식을 지닌 일반사용자 10명과 인공지능에 대해서 적정수준 이상의 지식을 가진 전문사용자 10명을 대상으로 하였다.

본 논문에서 실시한 실험은 인공지능 관련 사이트를 대상으로 하였다. 또한 웹에서 이용하고 있는 멀티미디어 데이터를 배제하고 일반 HTML 문서만을 대상으로 실험하였다. 실험에 이용된 웹가이드는 <표 3>의 키워드에 대해서 10,000개의 사이트를 기반 사례로 이용하였다.

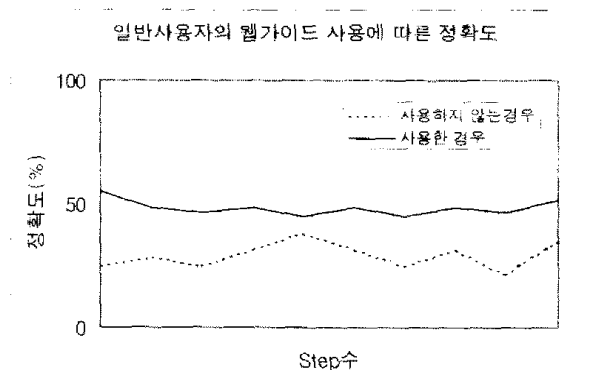
#### 4.2 평가

실험에 참가한 사용자들은 각각 <표 3>에서 제시한 키워드에 대해서 특정 사이트를 시작으로 자신의 관심 정도에 따라 원하는 사이트를 방문하여 방문하는 페이지

<표 3> 실험에 이용된 키워드 리스트  
 <Table 3> Keyword Lists for experiment

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

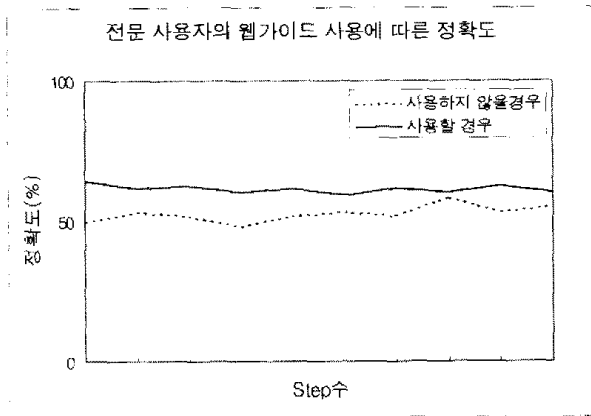
지에 대한 정확도를 평가하는 방법이다. 정확도는 실험에 참가한 사용자들이 검색된 결과에 대한 평가를 수치화 한 값으로, 사용자는 현재 페이지에 대해서 0 ~ 10까지의 정확도를 평가할 수 있다. 사용자가 한번의 링크를 클릭해서 다음 페이지로 이동하는 것을 스텝으로 정의하였으며 이전 페이지로 복귀하는 스텝 수는 인정하지 않았다. 또한 하나의 페이지에 대해서 웹가이드가 추천해주는 링크의 수는 최대 5개로 한정하였다. 본 실험에 사용된 평가함수는 태그의 가중치를 평가하는  $AV_k(p)$ 이다. 평가에 이용된 값은 매 스텝마다 실험에 참여한 사용자가 주어진 키워드에 대한 현재 페이지를 평가한 정확도의 평균값을 매 스텝에서의 최고점으로 나눈 값이다. 첫 번째 실험방법은 인공지능에 대한 특별한 지식이 없는 일반사용자를 대상으로 실험하였다. 사용자는 주어진 키워드에 대한 일반적인 정의를 지정된 스텝 동안 검색하며 이때 웹가이드를 사용할 경우와 그렇지 않을 경우에 따른 정확도를 비교하였다. (그림 9)는 실험에 따른 결과 그래프이다.



(그림 9) 일반사용자의 웹가이드 사용여부와 정확도 관계  
 (Fig. 9) Relation of exactness by common user(use or not Web-Guide)

(그림 9)의 결과와 같이 인공지능에 대한 일반적인 지식이 없는 일반사용자가 웹가이드를 사용하지 않은 경우에는 정확도가 비교적 낮았다. 웹가이드를 사용한 경우에는 그렇지 않은 것보다 전체적으로 정확도가 높았으며, 스텝 수의 변화에 따른 정확도의 변화폭이 감소하였다. 이는 사례베이스를 가지고 있는 웹가이드가 특정 사이트를 추천하는 경우 전문 분야에 특별한 전문 지식이 없는 일반 사용자들도 웹가이드가 자신이 원하는 사이트를 검색하는 데 상당한 도움을 주었음을 입증한다. 또한 첫 번째 스텝에서는 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 스텝 동안 검색하며 이때 웹가이드를 사용할 경우와 그렇지 않을 경우에 따른 정확도를 비교하였다. 사용된 평가함수와 정확도의 기준은 첫 번째 실험과 동일하다.



(그림 10) 전문 사용자의 웹가이드 사용여부와 정확도 관계 (Fig. 10) Relation of exactness by expert(use or not Web-Guide)

전문 사용자의 경우에도 웹가이드를 사용하였을 때의 정확도가 웹가이드를 사용하지 않았을 때보다 높은 결과를 나타내었다. 또한 정확도의 변화폭도 웹가이드를 사용하였을 때가 웹가이드를 사용하지 않았을 때보다 작은 수치로 나타났다. 그러나 첫 번째 실험에 비해서 웹가이드의 사용여부가 사용자의 정확도평가에는 적게 영향을 미쳤다. 이는 전문 사용자의 기반지식과 웹가이드의 사례베이스에 존재하는 지식의 공유부분이 많

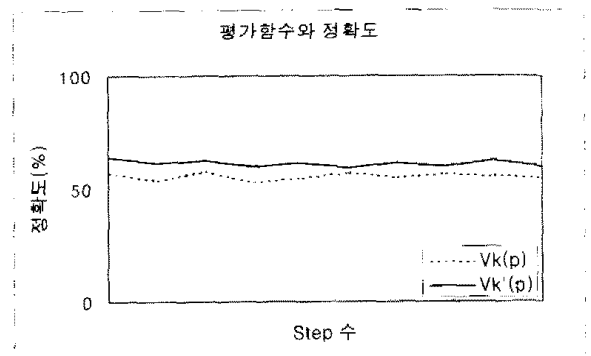
이 존재하기 때문이다. <표 4>는 각 실험에 대해서 사용자의 평균 정확도를 나타낸다.

<표 4> 사용자의 평균 정확도 (Table 4) Average exactness of users

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

평균 정확도는 실험에 참가한 사용자들의 정확도의 평균값을 나타낸 것이다. 실험 결과 랜덤의 경우 평균 18.6%의 정확도를 나타내었다. 일반 사용자가 웹가이드를 이용했을 경우에는 그렇지 않았을 경우보다 약 16.3% 정도의 정확도 향상이 있었으며, 전문 사용자의 경우에는 웹가이드를 이용했을 경우가 그렇지 않았을 경우보다 약 10.7% 정도의 정확도 향상이 있었다. 그러나 일반 사용자가 웹가이드의 추천을 이용하였을 경우보다 전문사용자가 자신의 지식을 바탕으로 검색하였을 경우에는 전문 사용자의 정확도가 우수한 것으로 평가되었다. 이는 웹가이드가 학습한 기반 지식이 전문 사용자의 기반 지식에는 이르지 못한 것을 보여준다.

다음으로 전문사용자가 10,000개의 기반사례를 바탕으로 웹가이드를 이용할 경우 논문에서 제시한 평가 함수에 따른 정확도를 비교하였다. 실험 방법과 평가 기준은 첫 번째 방법과 동일하다.



(그림 11) 평가 함수와 정확도 관계 (Fig. 11) Relation between evaluation function and exactness



평가 함수에 따른 사용자의 정확도는 현재 페이지에 나타나는 키워드의 빈도수만을 고려하는 평가 함수인  $AV_k(P)$ 보다 키워드에 이용된 태그를 조사하여 평가하는  $AV'_k(P)$ 가 우수하였다. 또한 정확도의 변화폭도  $AV_k(P)$ 보다  $AV'_k(P)$ 가 우수하였다. 이는 하나의 페이지를 설계하는 사람들이 자신이 강조하고자 하는 키워드에는 HTML의 특징인 태그를 많이 이용하며 태그의 특징을 분석한 경우 보다 정확한 사이트를 검색할 수 있었다. 다음은 각 평가 함수를 이용하였을 때의 평균 정확도를 나타낸다.

〈표 5〉 평가 함수와 평균 정확도  
 〈Table 5〉 Evaluation function and average exactness

평가 함수	평균 정확도(%)
$AV_k(p)$	55.7
$AV'_k(p)$	61.3

사용자의 정확도를 평균적으로 평가하면 키워드에 이용된 태그를 조사하여 평가하는  $AV'_k(P)$ 가 현재 페이지에 나타나는 키워드의 빈도수만을 고려하는 평가 함수인  $AV_k(P)$ 보다 약 5.6% 정도의 정확도가 향상되었다.

### 5. 결 론

인터넷에 존재하는 많은 정보 중에서 사용자 자신이 원하는 정보를 찾고자 하는 연구가 활발히 진행되었다. 그러나 검색엔진과 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 에이전트에 대해서 연구하였다. 제안한 웹가이드는 특정 개인의 관심 분야를 지속적으로 관찰, 학습하는 에이전트로 개인의 관심분야만을 집중적으로 학습한다. 웹가이드는 인터넷 시스템과 학습 시스템의 두 개의 부시스템으로 이루어져 있다. 하나의 HTML 문서를 표현하기 위해서 하나의 문서에 나타난 단어를 벡터(Vector) 방식으로 표현하는 TFIDF-Vector 방식에 태그를 추가하여 검사하는 방식을 이용하였다. 문서의 가중치를 평가하는 정보 추출방식으로는 사용자가 입력한 키워드가

문서에 나타나는 빈도수를 평가하는 방식과 키워드에 이용된 태그에 가중치를 부여하여 평가하는 방식을 이용하였다. 학습 방법으로는 평가 함수를 통해 평가된 사례를 기반으로 한 사례기반 학습을 이용하였다. 실험 결과 일반사용자와 전문사용자 모두 자신이 원하는 정보를 웹가이드를 이용하였을 경우가 그렇지 않았을 경우보다 빠르게 얻을 수 있었다. 또한 키워드에 가중치를 두어 평가한 평가함수를 이용한 웹가이드가 그렇지 않은 경우보다 좋은 결과를 얻었다. 결과적으로 웹가이드를 이용하면 사용자는 자신만의 관심 분야에 대해서 보다 빠른 시간내에 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

- [1] Armstrong, R., Freitag, D., Joachims, T., Mitchell, T., "WebWatcher: A Learning Apprentice for the World Wide Web", *AAAI 1195 Spring Symposium on Information Gathering From Heterogeneous, Distributed Environments*, Stanford, March 1995.
- [2] Dunja Mladenic, "Personal WebWatcher: design and implementation", *Carnegie Mellon University*, 1996.
- [3] Marko Balabanovic, Yoav Shoham, "Learning Information Retrieval Agents: Experiments with Automated Web Browsing", *Stanford University*, 1996.
- [4] Thorsten Joachims, Tom Mitchell, Dayne Freitag, Robert Armstrong, "WebWatcher: Machine Learning and Hypertext", *Carnegie Mellon University*, 1995.
- [5] Henry Lieberman, "Letizia: An Agent That Assists Web Browsing", *Massachusetts Institute of Technology*, 1996.
- [6] Claire Cardie, "A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis", *Department of Computer Science University of Massachusetts Amher-*

st, MA 01003, 1996.

- [7] 최중민, "적용형 에이전트", *정보과학회지 제15권 제3호*, pp.29-37, 1997.3.
- [8] David Abdu and Oren Bar-Ner, "Software Agents: A General Overview", 1996.
- [9] Robert C. Holte, Chris Drummond, "A Learning Apprentice For Browsing", *University of Ottawa*, 1995.
- [10] Chris Drummond, Robert Holte, Dan Ionescu, "Accelerating Browsing by Automatically Inferring a User's Search Goal", *Computer Science Dept. University of Ottawa, Ottawa, Ontario, Canada. K1N 6N5*, 1995.
- [11] M. Pazzani, L. Nguyen, S. Mantik, "Learning from hotlists and coldlists: Towards a WWW information filtering and seeking agent", 1996.
- [12] T.M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, "Experience with a Learning Personal Assistant", *Communications of the ACM, Vol.37, No.7*, pp. 81-91, July 1994.



**김 영 권**

1987년 동국대학교 수학과 졸업 (이학사)  
 1993년 중앙대학교 컴퓨터공학과 (공학석사)  
 1994년~현재 중앙대학교 컴퓨터 공학과(박사과정)

관심분야 : 퍼지 시스템, 신경망 시스템, 사례기반 추론 등



**이 재 필**

1993년 중앙대학교 전자계산학과 졸업(이학사)  
 1995년 중앙대학교 컴퓨터공학과 (공학석사)  
 1995년~현재 중앙대학교 컴퓨터 공학과(박사과정)

관심분야 : 사례기반 추론, 데이터 마이닝 등



**이 말 레**

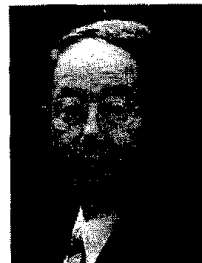
1991년 국립군산대학교 전자계산 학과 졸업(학사)  
 1993년 중앙대학교 대학원 컴퓨터공학과(공학석사)  
 1997년~현재 중앙대학교 대학원 컴퓨터공학과 박사수료

관심분야 : 인공지능, 유전자 알고리즘, 퍼지 이론, 신경망, 인공생명



**김 경 만**

1996년 중앙대학교 컴퓨터공학과 졸업(이학사)  
 1998년 중앙대학교 컴퓨터공학과 졸업(공학석사)  
 1998년~현재 온넷 선임 연구원  
 관심분야 : 에이전트 시스템, 정보추출, 기계학습, 사례기반 추론 등



**김 기 태**

1970년 중앙대학교 수학과 응용수학(이학석사)  
 1988년 숭실대학교 전자계산학과 (공학박사)  
 1972년 중앙대학교 전자계산학과 강사

1986년 12월~1987년 2월 일본 동경공업대학 연구교수  
 1991년 1월~1992년 1월 미국 Connecticut 주립대학 객원교수

1979년~현재 중앙대학교 컴퓨터공학과 교수  
 관심분야 : 컴파일러, 알고리즘, 전문가 시스템, 신경망 시스템 등