

# 학습 방법을 이용한 지능형 웹 에이전트 시스템 설계

## Design Intelligent Web-Agent System Using Learning Method

이말례(Mal-Rey Lee)\*, 남태우(Tae-Woo Nam)\*\*

### 목 차

- |                   |        |
|-------------------|--------|
| 1. 서 론            | 4. 구 현 |
| 2. 기반 연구          | 5. 결 론 |
| 3. 지능형 웹 에이전트 시스템 |        |

### 초 록

많은 양의 정보가 인터넷을 통해 제공되고 있다. 이 때문에 사용자는 쓸모없는 정보를 찾는 경우가 많다. 본 논문에서는 이와 같은 사용자의 불편함을 해결하기 위하여 지능형 웹 에이전트 시스템을 제

안한다. 이 지능형 웹 에이전트 시스템은 사용자의 행동과 에이전트 방문을 키워드를 중심으로 각각의 사례로 저장하는 사례 기반 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. 지능형 웹 에이전트 시스템은 인터페이스 시스템과 학습 시스템의 두 개의 부시스템으로 이루어져 있다. 실험 결과 지능형 웹 에이전트 시스템을 사용했을 때가 사용하지 않았을 때보다 훨씬 빨리 찾을 수 있었다.

### ABSTRACT

Massive amount of information is provided for the internet users. Therefore, the users are exposed even to the useless information. In this paper, a Intelligent Web-Agent system is present as a solution for this kind if users inconvenience. This Intelligent Web-Agent system is devised users to search by the keyword about which they get information and commend the sites which have more intensive relation with the examine keyword, judge by the users and the case-base constructed by the Intelligent Web-Agent system itself previously, so the users can access the essential web sites in short time. Intelligent Web-Agent system is compose of a interface-system and a learning system. According to the experiment, using the Intelligent Web-Agent System quicker than the case when not using the Intelligent Web-Agent System.

\* 중앙대학교 컴퓨터공학과 강사

\*\* 중앙대학교 문헌정보학과 교수

■ 논문 접수일 : 1997년 12월 3일

## 1. 서론

일반적으로 사용자는 보다 쉽게 원하는 자료를 검색하기 위해서 사이트에 대한 정보를 저장하고 있는 검색엔진을 이용하는 경우가 대부분이다. 하지만 자료의 양이 방대해짐에 따라 검색엔진을 통한 결과물도 사용자들에게는 필요 이상의 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 검색해 보면 약 30만건이 결과물로 제시된다. 또한 검색엔진은 불특정 다수를 대상으로 하기 때문에 하나의 키워드에 대해서 사용하는 사람에 따라 전혀 다른 결과물을 얻을 수 있다.

본 논문에서는 이러한 문제를 해결할 수 있는 지능형 웹 에이전트 시스템을 제안한다. 지능형 웹 에이전트 시스템은 사용자의 행동과 에이전트의 방문을 키워드를 중심으로 각각의 사례로 저장하는 사례 기반 학습 방법을 이용하여 특정 개인 사용자가 웹상에서 검색하고자 하는 자료를 입력받은 후부터 사용자의 방문 행동을 학습하여 보다 빠른 시간내에 원하고자 하는 자료를 검색할 수 있도록 도와주는 에이전트 시스템이다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 기반연구로 이용된 에이전트와 사례 기반 학습에 대하여 설명한다. 또한 기존에 연구되어 온 관련연구를 고찰한다. 3장에서는 본 논문에서 제안한 지능형 웹 에이전트 시스템에 대하여 설명한다. 지능형 웹 에이전트 시

스템의 개요와 전체적인 시스템 구조, 이 시스템에서 사용하는 문서표현방법, 정보추출방법 등에 대하여 설명한다. 4장에서는 지능형 웹 에이전트 시스템 구현에 대해서 설명한다. 5장에서는 본 논문에 대한 결론을 맺는다.

## 2. 기반 연구

2장에서는 본논문의 기반연구로 사용된 에이전트와 사례 기반 학습에 대하여 설명하며 본 논문의 관련연구로 사용된 연구들에 대하여 고찰한다.

### 2.1 에이전트

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.

#### 2.1.1 정의

에이전트는 바라보는 시각과 연구목적에 따라 여러 가지 매우 다양한 정의를 가질 수 있다. 에이전트에 대한 일반적인 정의는 다음과 같다(최중민 1997, 8-10).

\* 에이전트는 특정 목적에 대하여 사용자를

대신하여 작업을 수행하는 자율적 프로세스(autonomous process)이다.

\* 에이전트는 독자적으로 존재하지 않고 어떤 환경의 일부이거나 그 안에서 동작하는 시스템이다. 여기서의 환경은 운영체제, 네트워크, 또는 MUD 게임환경 등을 지칭한다.

\* 에이전트는 지식베이스와 추론 기능을 가지며 사용자, 자원(resource), 또는 다른 에이전트와의 정보교환과 통신을 통해 문제 해결을 도모한다.

\* 에이전트는 스스로 환경의 변화를 인지하고 그에 대응하는 행동을 취하며, 경험을 바탕으로 학습하는 기능을 가진다.

정리하면 에이전트는 스스로 여러 가지 복합적인 상황을 판단하여 학습하며, 이를 지식베이스에 저장할 수 있어야 한다. 에이전트는 환경의 변화를 감지하여 스스로 행동을 취할 수 있다. 또한 에이전트는 다른 에이전트와의 통신을 통하여 새로운 지식을 학습할 수 있으며 지속적인 수행을 통해서 환경의 변화를 추구할 수 있어야 한다.

에이전트는 위의 정의를 모두 만족시킬 수도 있지만 각각의 에이전트의 종류에 따라서 특정 부분만을 만족할 수도 있다.

### 2.1.2 특징

이러한 에이전트는 일반적으로 다음과 같은

특징을 지니고 있다(최중민 1997).

#### (1) 자율성

에이전트가 다른 프로그램과 구별할 수 있는 가장 다른 특징으로 사용자의 특별한 지시가 없어도 에이전트는 스스로 판단하여 행동하는 특징을 말한다.

#### (2) 학습성

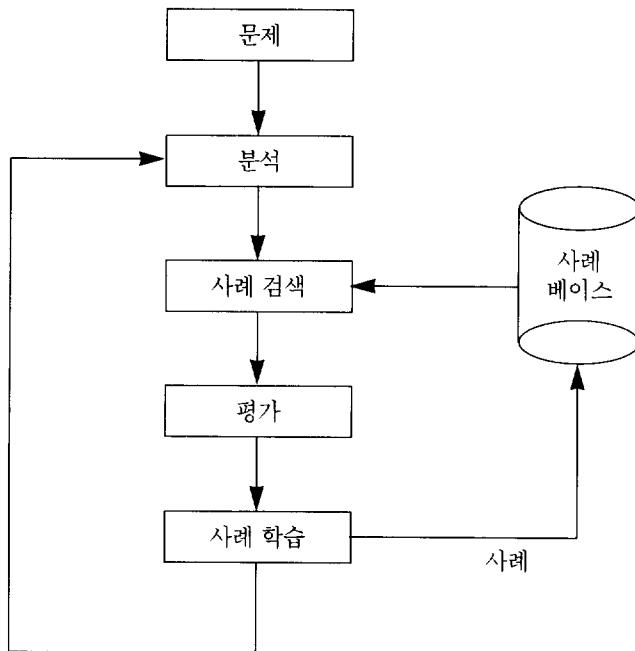
학습성은 에이전트 스스로가 특정 문제에 대한 해를 파악하여 자신의 능력을 강화시키는 특징을 말한다. 학습을 위한 알고리즘은 이미 인공지능에서 많이 연구된 기법을 기반으로 수행된다.

#### (3) 이동성

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.

## 2.2 사례 기반 학습

사례 기반 추론은 과거의 특정 문제에 대한 해결방안으로 사용되었던 해를 사례로 저장하며 새로운 문제에 대한 해를 추론하기 위해서 과거의 사례를 이용하는 방법이다. 사례 기반 학습은 사례 기반 추론과는 달리 적응성이 아닌 학습에 중점을 두고 있다. 사례 기반 학습



〈그림 1〉 사례 기반 학습의 일반적인 방법

에서는 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(이재필 & 양경직 1995).

사례 기반 학습은 일반적으로 〈그림 1〉의 절차에 따라 이루어진다.

### 2.3 관련연구

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. 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 이

를 기준으로 다음 사용자에게 추천해주는 방식을 이용한다. WebWatcher 에서는 문서를 표현하는 방식으로 하나의 문서를 단어단위로 분류하여 Boolean 값을 갖는 bag-of-words 방식을 이용하였고 평가함수로는 TFIDF를 이용하였다. 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 (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 방식은 Web-Watcher 와 동일하며 평가함수는 Bayesian Classifier 방식을 이용하였다. 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 (Dunja 1996).

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

본 논문은 이러한 문제점을 해결하기 위하여 사용자와 에이전트의 사례를 복합적으로 평가하는 사례 기반학습을 이용하였으며, 학습기간의 단축 및 보다 정확한 사이트의 추천

〈표 1〉 기존 연구와의 비교

	Web-Watcher (Armstrong 96)	Personal Web-Watcher (Dunja 96)	LIRA (Marko 96)	본 논문
대상 사용자	불특정 다수	개인	개인	개인
이용방법	검색	검색	써핑	검색
문서표현방법	bag-of-words (Boolean)	bag-of-words (Boolean)	Vector	bag-of-words (frequency)
정보추출	TFIDF	Bayesian classifier	relevance feedback	태그 평가
학습 기준	키워드	사용자의 행동	사용자의 행동	키워드

을 위하여 사용자의 대상을 특정개인으로 한정하여 키워드의 관심여부를 보다 명확히 하였다. 또한 하나의 문서를 표현하기 위해서 Web-Watcher 와 동일한 방식인 bag-of-words 방식을 이용하였으나 문서의 평가값을 순차적으로 표현할 수 있는 방식을 채택하였으며 키워드에 이용된 태그를 복합적으로 평가하는 평가함수를 채택하였다. <표 1>은 본 연구과 기존연구를 비교하였다.

학습하여 다음 사이트를 추천해주는 개인형 웹 에이전트 시스템이다. 지능형 웹 에이전트 시스템은 사용자가 입력한 키워드를 중심으로 한 사례 기반 학습을 이용하며 개인의 사례가 자신의 시스템에 존재하는 클라이언트 기반의 시스템이다. 지능형 웹 에이전트 시스템은 사용자와의 인터페이스를 담당하는 인터페이스 시스템 (Interface System)과 사용자의 방문 사이트를 학습하는 학습 시스템 (Learning System)으로 구성된다.

### 3. 지능형 웹 에이전트 시스템

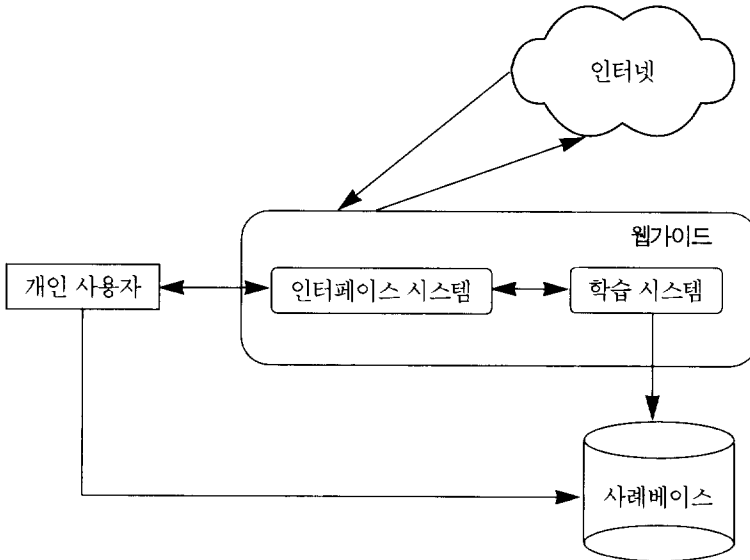
#### 3.1 개요

본 논문에서 제안하고 있는 지능형 웹 에이전트 시스템은 인터넷에서 개인의 관심분야를

#### 3.2 구조

지능형 웹 에이전트 시스템의 구조는 <그림 2>와 같다.

인터페이스 시스템은 사용자와 지능형 웹 에이전트 시스템 사이의 모든 인터페이스 작



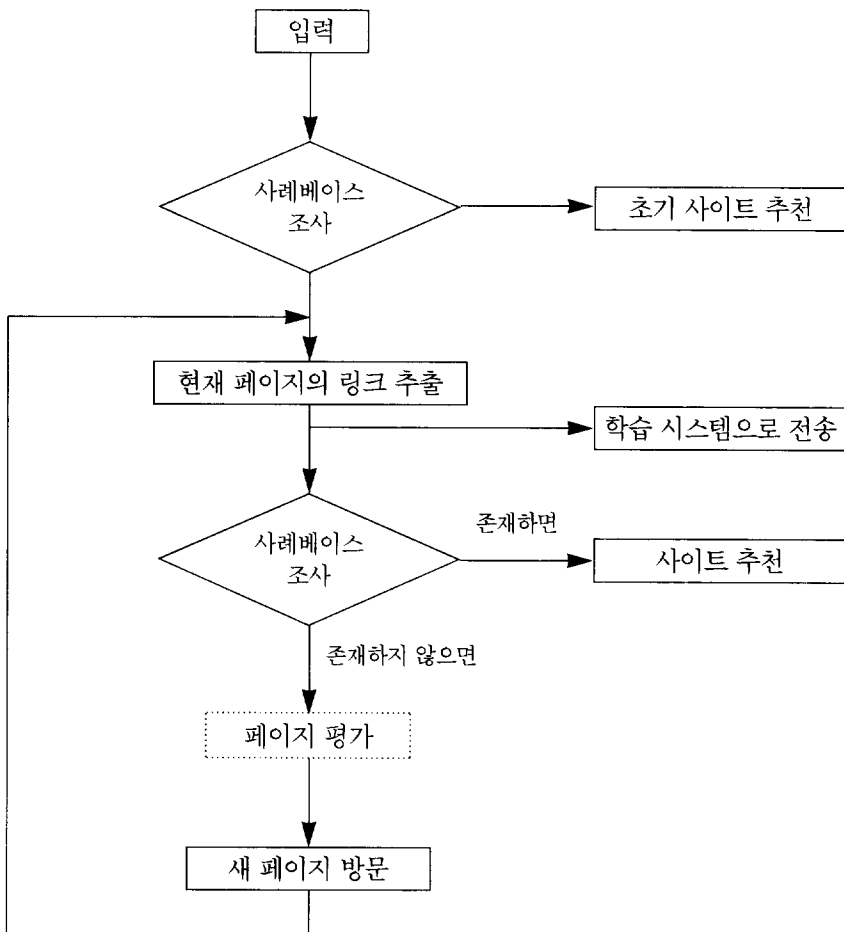
<그림 2> 지능형 웹 에이전트 시스템

업을 담당한다. 사용자는 인터페이스 시스템을 통해서 웹브라우저와 같은 기본적인 작업을 수행할 수 있으며 학습 시스템을 통해서 나온 추천사이트등을 검색할 수 있다. 또한 현재 사용자가 방문중인 사이트에 대해서 사용자 스스로가 바로 평가하여 사례베이스를 갱신할 수 있다. 학습 시스템은 인터페이스 시스템과의 통신을 통해서 사용자가 방문한 사이트들에 대해서 각각 새로운 사례들을 학습 알고리

즘을 통해 학습하여 사례베이스를 갱신하는 작업을 수행한다.

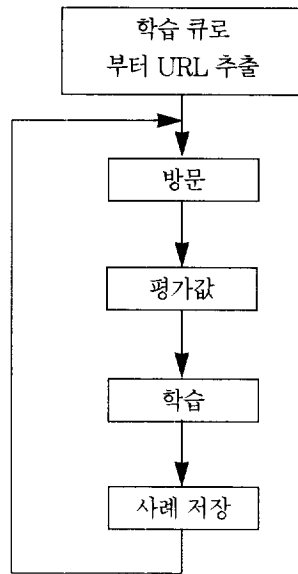
### 3.2.1 인터페이스 시스템

인터페이스 시스템의 수행과정은 <그림 3>과 같다. 인터페이스 시스템에서는 먼저 사용자로부터 자신이 관심있어 하는 키워드를 입력받는다. 사용자로 키워드를 입력받은 지능



<그림 3> 인터페이스 시스템

형 웹 에이전트 시스템은 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.



〈그림 4〉 학습 시스템

### 3.2.2 학습 시스템

학습 시스템의 수행과정은 〈그림 4〉과 같다.

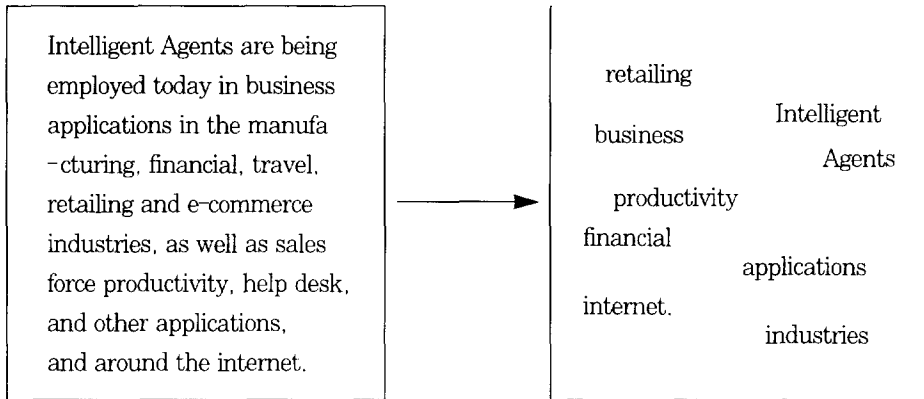
학습 시스템은 인터페이스 시스템으로부터 전송받은 자료가 저장되어 있는 학습큐 (Learning Queue)로부터 URL을 하나씩 추출하여 차례로 방문한다. 학습 시스템은 방문한 URL로부터 전송받은 HTML 문서에 대해서 평가값을 평가 함수를 통해 결정한 후 학습 알고리즘에 따라 사례에 저장한다. 이러한 작업은 학습큐(Learning Queue)에 링크가 없어질 때까지 반복하며 수행된다. 학습 시스템에서 수행되는 작업은 수행 속도의 향상을 위해서 네트워크의 사용빈도가 떨어지는 야간작업에 주로 수행한다.

### 3.3 문서 표현 방법

일반적으로 정보검색이나 텍스트 학습에서 하나의 문서를 표현하기 위해서는 TFIDF-vector 표현 방식을 많이 이용한다. TFIDF-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 HTML 과 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





〈그림 5〉 일반적인 TFIDF-Vector 표현방식

태그들을 포함하고 있다. HTML 문서에서 사용하는 태그들은 “<” 로 시작하며 “/” 로 종료하는 구조를 가지고 있다. 그러므로 지능형 웹 에이전트 시스템은 각 단어가 포함하고 있는 태그까지 하나의 단어로 분류한다.

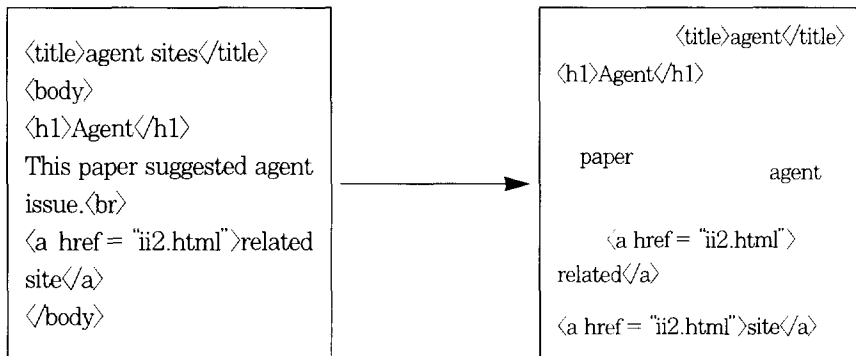
이런 하나의 문서 d에서 단어 w에 대한 weight 값을 산출하는 방식으로 다음의 수식으로 표현할 수 있다(Armstrong 1996).

### 3.4 정보 추출 방법

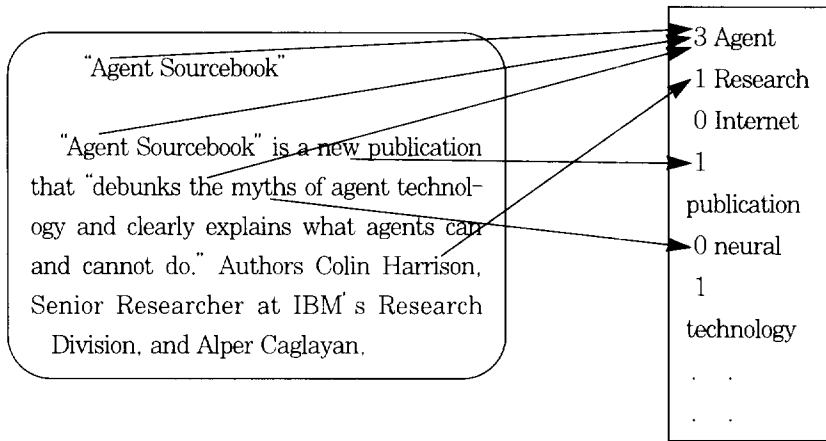
하나의 문서에서 정보를 추출하는 방법으로 TFIDF 방식을 많이 사용한다. TFIDF 방식

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

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



〈그림 6〉 본 논문에 이용된 문서 표현 방식



〈그림 7〉 일반적인 TFIDF 방식

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

지능형 웹 에이전트 시스템은 보다 정확한 정보 수집을 위하여 사용자로부터 관심 키워드를 직접 입력받는 방식을 이용한다. 지능형 웹 에이전트 시스템은 사용자로부터 입력받은 키워드를 중심으로 하여 정보를 추출한다. 지능형 웹 에이전트 시스템은 사용자가 현재 방문하고 있는 홈페이지에 있는 링크들을 미리 방문하여 각 페이지에 대해서 현재 사용자의 관심분야에 대해서 정보를 추출한다.

지능형 웹 에이전트 시스템은 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 다음

$$AV_k(P) = TF(P)$$

TF(P) : 특정 페이지 P에 키워드 k가 나타나는 빈도수

과 같이 정의된다.

두 번째 방법은 첫 번째 방식에 HTML 문서의 특징을 검사하는 방법을 추가하는 방식이다. 일반적으로 HTML 문서를 제작하는 사람들은 자신이 외부에 알리고자 하는 내용에 대해서는 여러 가지 태그를 이용하는 방식을 많이 이용한다. 지능형 웹 에이전트 시스템은 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.

$$AV\ k(P) = \sum TV(P)$$

TV(P) : 특정 페이지 P에서 키워드 k가 태그 t를 사용할 때의 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 2>의 가중치를 부여하였다.

<표 2> 태그에 따른 가중치

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

### 3.5 사례 기반 학습 알고리즘

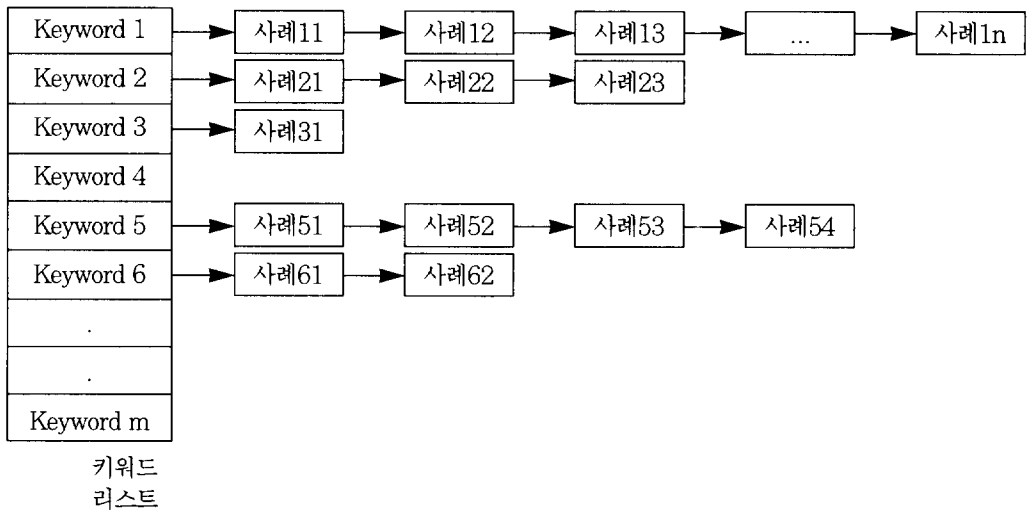
지능형 웹 에이전트 시스템에서 사용하는 학습 알고리즘은 특정 개인 사용자로부터 관심 키워드를 입력받아 하나의 사례로 저장, 평가하여 학습하는 방법이다. 지능형 웹 에이전트 시스템은 지능형 웹 에이전트와 사용자가 방문한 모든 홈페이지를 하나의 사례로 간주한다. 하나의 사례에는 <표 3>의 내용이 저장된다.

<표 3> 사례의 저장 내용

내용	비고
ID	사례의 고유번호
URL	사이트 주소 저장
Agent	웹가이드의 평가값 저장
Value	사용자의 평가값 저장

사용자는 자신이 검색하고자 하는 키워드를 입력할 수 있으며 지능형 웹 에이전트 시스템은 사용자로부터 입력받은 키워드를 인덱스화 하는 리스트를 구성한다. 지능형 웹 에이전트 시스템에서 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.

하나의 사례는 <그림 8>과 같이 키워드를 중심으로 이루어지며 평가함수 값에 따라 정렬되어 저장된다.



〈그림 8〉 사례베이스 표현 방법

지능형 웹 에이전트 시스템은 인터페이스 시스템으로부터 전송받은 자료들은 학습시스템의 Learning Queue 에 저장된다. 학습시스템은 Learning Queue에 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 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 평가값이 우수한 사례를 수정한다.

〈그림 9〉는 지능형 웹 에이전트의 학습 알고리즘이다.

또한 지능형 웹 에이전트 시스템은 특정 개인을 대상으로 하는 시스템이므로 사용자의 평가가 학습에 중요한 영향을 미치게 된다. 사용자는 자신이 현재 방문하고 있는 임의의 페이지에 대해서 평가를 할 수 있으며 지능형 웹 에이전트 시스템은 사용자의 평가에 따라 현재 페이지에 대한 사례를 수정한다. 현재 페이지에 대한 사용자의 평가를 학습하는 알고리즘은 〈그림 10〉과 같다.

```

WHILE Get_Case() NOT empty
  Get_Case() : // 새로운 사례를 추출한다.
  IF Exist_Keywordlist then
    // 키워드리스트에 존재하면
    Check_Case() : // 새로운 사례의 존재여부 확인
    // 사례가 이미 존재하면
    Check_Value() : // 기존의 사례의 평가값을 검사하여
    IF Now_Case_Value > Exist_Case_Value then
      // 새로운 사례의 평가값이 크면
      Update_Case() : // 사례의 평가값을 수정한다.
    End IF
  ELSE // 사례가 존재하지 않으면
    Insert_Case() : // 새로운 사례로 입력
  End IF
LOOP // 새로운 사례가 없을 때까지 반복
    
```

〈그림 9〉 웹가이드 학습 알고리즘

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

#### 4. 구현

지능형 웹 에이전트 시스템은 윈도우즈 95 환경에서 인터넷 사용이 가능한 IBM-PC에서 비주얼 베이직 5.0 으로 구현하였다. 지능형

```

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

〈그림 10〉 사용자에게 의한 사례학습 알고리즘

웹 에이전트 시스템은 인터페이스 시스템과 학습 시스템의 두가지 부시스템으로 구현하였다. 실제 사용자는 인터 페이스 시스템만을 이용하게 된다. 지능형 웹 에이전트 시스템의 사용방식은 다음과 같다.

#### 4.1 인터페이스 시스템

인터페이스 시스템은 기존의 웹 브라우저와 동일한 환경을 제공하도록 구현하였으며 인터 페이스 시스템은 사용자에게 현재 링크 중 사용자의 관심 키워드에 대한 평가 값이 높은 순서대로 추천해 준다. 인터페이스 시스템의 운영 방식은 다음과 같다. <그림 11>은 지능형 웹 에이전트 인터페이스 시스템의 주 화면이다. 지능형 웹 에이전트 시스템은 현재 보여지고 있는 페이지의 모든 링크에 대해서 사례베

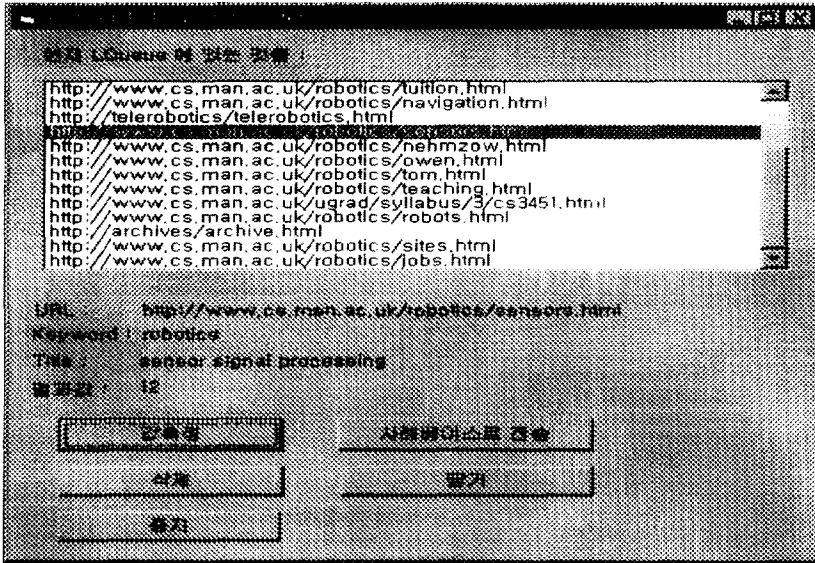
이스와 비교하여 임계값이 높은 순서대로 특정 링크를 화면 윗 부분에 추천 싸이트로 표시한다. 그리고 현재 페이지의 링크를 학습시스템으로 전송한다. 사용자는 자신이 원하는 링크를 클릭할 수 있으며 운영방식은 기존의 웹 브라우저와 동일하다. 또한 사용자는 인터페이스 시스템을 통해서 현재 페이지에 대해서 직접적으로 평가를 할 수 있다.

#### 4.2 학습 시스템

학습 시스템은 <그림 12>와 같으며 하나의 프로그램으로 구성되어 있다. 사용자는 학습 시스템을 통해서 현재 학습큐(Learning Queue)에 저장되어 있는 내용을 확인할 수 있으며 실시간으로 특정 페이지에 대한 평가 값을 확인할 수도 있다. 또한 존재하지 않는



<그림 11> 지능형 웹 에이전트 인터페이스 시스템 화면



〈그림 12〉 지능형 웹 에이전트 학습 시스템 화면

사이트는 삭제할 수 있다.

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

## 5. 결론

인터넷에 존재하는 많은 정보 중에서 사용자 자신이 원하는 정보를 찾고자 하는 연구가 활발히 진행되었다. 하지만 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. 본 논문에서 제안하고 있는 지능형 웹 에이전트 시스템은 특정 개인의 관심 분야를 지속적으로 관찰, 학습하는 에이전트로 개인의 관심분야만을 집중적으로 학습한다. 지능형 웹 에이전트 시스템은 인터페이스 시스템과 학습 시스템의 두 개의 부시스템으로 이루어져 있다. 하나의 HTML 문서를 표현하기 위해서 하나의 문서에 나타난 단어를 Vector 방식으로 표현하는 TFIDF-Vector 방식에 태그를 추가하여 검사하는 방식을 이용하였다. 문서의 가중치를 평가하는 정보추출방식으로는 사용자가 입력한 키워드가 문서에 나타나는 빈도수를 평가하는 방식과 키워드에 이용된 태그에 가중치를 부여하여 평가하는 방식을 이용하였다. 학습 방법으로는 평가함수를 통해 평가된 사

례를 기반으로 한 사례 기반 학습을 이용하였다. 실험 결과 일반사용자와 전문사용자 모두 자신이 원하는 정보를 지능형 웹 에이전트 시스템을 이용하였을 경우가 그렇지 않았을 경우보다 빠르게 얻을 수 있었다. 또한 키워드에 가중치를 두어 평가한 평가함수를 이용한 지

능형 웹 에이전트 시스템이 그렇지 않은 경우보다 좋은 결과를 얻었다. 결과적으로 지능형 웹 에이전트 시스템을 이용하면 사용자는 자신만의 관심 분야에 대해서 보다 빠른 시간내에 원하는 결과를 얻을 수 있게 된다.

## 참고문헌

- 최중민, 1997. "에이전트의 개요와 연구방향", *정보과학회지 제15권 제3호*, 8-10.
- 최중민, 1997. "적응형 에이전트", *정보과학회지 제15권 제3호*, 29-37.
- 이재필, 1995. "규칙기반 추론과 사례기반 추론의 결합모델", 석사학위논문, 중앙대학교 전자계산학과.
- 양경직, 1995. "탐색 시간이 제한되는 경우의 개선된 사례 기반 탐색 알고리즘", 석사학위논문, 중앙대학교 전자계산학과, 1995.
- 이대령 & 박영택, 1996. "정보추출을 이용한 기계학습기반의 웹에이전트", 한국정보과학회 춘계 학술발표논문집, 201-204.
- Armstrong, R., Freitag, D., Joachims, T. 1996. "WebWatcher : A Learning Apprentice for the World Wide Web", *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March.
- Dunja Mladenic, 1996. "Personal Web-Watcher : design and implementation", *Carnegie Mellon University*.
- Marko Balabanovic, Yoav Shoham, 1996. "Learning Information Retrieval Agents : Experiments with Automated Web Browsing", *Stanford University*.
- Henry Lieberman, 1996. "Letizia : An Agent That Assists Web Browsing", *Massachusetts Institute of Technology*.
- Thorsten Joachims, Tom Mitchell, Dayne Freitag, Robert Armstrong, 1995. "WebWatcher : Machine Learning and Hypertext", *Carnegie Mellon University*.
- Robert C. Holte, Chris Drummond, 1995. "A Learning Apprentice For Browsing", *University of Ottawa*.
- Henry Lieberman, 1996. "Autonomous Interface Agents", *Massachusetts Institute of Technology*.



- David Abdu and Oren Bar-Ner, 1996. "Software Agents : A General Overview".
- Patrick Henry Winston, 1984. "Artificial Intelligence Second Edition", *Addison-Wesley*.
- Elaine Rich, Kevin Kinghtm, 1991. "Artificial Intelligence Second Edition", *McGraw-Hill*.
- George F.Luger, William A. Stubblefield, 1993. "Artificial Intelligence structures and strategies for complex problem solving Second Edtion", *Benjamin/Cummings Publishing Company*, 133- 137.
- Claire Cardie, 1996. "A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis", *Department of Computer Science University of Massachusetts Amherst, MA 01003*.
- Chris Drummond, Robert Holte, Dan Ionescu, 1995. "Accelerating Browsing by Automatically Inferring a User's Search Goal", *Computer Science Dept. University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5*.
- M. Pazzani, L. Nguyen, S. Mantik, 1996. "Learning from hotlists and coldlists : Towards a WWW information filtering and seeking agent".
- T.M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, 1994. "Experience with a Learning Personal Assistant", *Communications of the ACM, Vol. 37, No. 7*, 81-91.
- Chung T.Kwok, Daniel S. Weld, 1996. "Planning to Gather Information", *Department of Computer Science & Engineering University of Washington*.
- Xiaobin Fu, Kristian J. Hammond, Robin Burke, 1996. "ECHO : An Information Gathering Agent", *The University of Chicago Computer Science Department*.