분산 정보 검색을 위한 신경망 기반 사용자 피드백 학습 메카니즘

최용석†

요 약

웹과 같은 분산 정보 검색 환경에서 문서들은 많은 문서 데이터 베이스들에 자연스럽게 분할되어서 존재한다. 그러므로 이러한 문서들의 효율적인 검색을 위해서는 먼저 질의에 관련되는 문서들을 제공할 것으로 판단되는 문서 데이터 베이스를 찾아내고 다음으로 그 문서 데이터 베이스에 질의를 줌으로써 분산 정보 검색을 수행해야 한다. 본 논문에서는 이러한 효율적인 분산 정보 검색을 위한 신경망 기반 사용자 피드백 학습 메카니즘을 제안한다. 제안된 학습 메카니즘은 질의 검색 예제들을 통하여 얻어진 질의에 대한 사용자의 관련도 피드백 정보에 기반하여 역전파 알고리즘으로 분산 정보 검색 지식을 학습한다. 충분히 학습한 후의 학습 메카니즘은 주어진 질의에 대하여 관련 문서 데이터 베이스들을 찾아내고 그 문서 데이터 베이스들로부터 관련되는 문서들을 검색하는데 효과적으로 사용될 수 있다. 실험에서는 제안된 학습 메카니즘을 적용한 신경망 에이전트 시스템을 구현하고 이 시스템의 정보 검색 성능을 기존 시스템들과 비교하여 제안된 학습 메카니즘의 유용성을 입증한다.

Neural Net Based User Feedback Learning Mechanism for Distributed Information Retrieval

Yong S. Choi†

ABSTRACT

Since documents on the Web are naturally partitioned into many document databases, the efficient information retrieval process requires identifying the document databases that are most likely to provide relevant documents to the query and then querying the identified document databases. We propose a neural net based user feedback learning mechanism for such an efficient information retrieval. Presented learning mechanism learns about underlying document databases using the relevance feedbacks obtained from user's retrieval experiences. For a given query, the learning mechanism, which is sufficiently trained, discovers the document databases associated with the relevant documents and retrieves those documents effectively.

1. 개요

사용자가 원하는 작업을 자동적으로 해결·처리하여 주는 다양한 형태의 프로그램들이 정보 기술의 많은 영역에서 에이전트라는 이름으로 개발되어져 왔다. 특히, 웹 상에 분산된 다양한 문서들이 급속히 증가함에 따라 사용자에게 원하는 문서를 찾아주는 정보 검색 에이전트에 대한 요구가 증대되어 왔고, 그 결과로 많은 웹 정보 검색 에이전트들이 개발되

† 정 회 원: 한양대학교 컴퓨터교육과 전임교수
논문접수: 2001년 6월 23일 심사완료: 2001년 8월 28일

어졌다.

검색 엔진이라고 불리는 전통적인 웹 정보 검색 에이전트는 일반적으로 각 문서에 대한 색인을 단일 플랫폼에 데이터 베이스 형태로 구축하고 TF×IDF 알고리즘에 기반한 벡터 공간 모델[1] 과 같은 전통적인 모델을 사용한다. 이러한 모델을 기반으로 하여 검색 엔진은 사용자 질의에 대한 각 문서의 관련도를 질의에 존재하는 용어가 문서 안에서 나타나는 횟수에는 비례하고 그 용어가 나타나는 전체 문서들의 개수에는 반비례하는 방식으로 계산하여 관련도가 높은 문서들을 순서대로 반환하게 된다.

그러나 이러한 방법은 Lycos[2] 나 WebCrawler [3]와 같은 검색 엔진에서와 같이 많은 사용자들에 의해서 매우 빈번히 검색이 이루어지는 환경에서는 심한 병목 현상과 접근 지연 현상을 일으킬 수 있다. 이러한 단점은 단일 플랫폼에 구축된 색인 데이터 베이스의 규모가 커질수록 더 심각해진다. 또한 웹과 같이 문서의 생성, 소멸, 변경 등이 비동기적으로 빈번하게 이루어지는 동적인 대규모 분산 환경에서 모든 문서들에 대한 색인들을 하나의 플랫폼에 일괄적이면서도 효과적으로 유지하는 것은 사실상 불가능하다.

메타 검색 엔진이라고 불리는 분산 웹 검색 에이전트는 이러한 문제를 해결하기 위하여 정보 검색 작업을 여러 개의 문서 데이터 베이스 들에 분산시키는 방법을 사용한다. 문서 데이터 베이스는 문서 모임(document collection) 과 임의의 질의에 대하여 그 문서 모임으로부터 관련된 문서를 반환해주는 색인 시스템으로 구성된 정보 자원을 뜻한다. 그러므로 일반적으로 사용되는 주제별 검색 엔진과 논문 검색을 위해서 사용되는 문헌 정보 검색 서비스 등은 모두 문서 데이터 베이스이다. 이와 같이 대규모의 문서들이 분산된 여러 문서 데이터 베이스들에 분할되어져 있는 환경에서 수행하는 정보 검색을 분산 정보 검색이라 한다. 이러한 분산 정보 검색을 수행하는 메타 검색 엔진으로는 NCSTRL[4], IBM InfoMarket[5], MetaCrawler[6] 등이 있으며 이들은 사용자의 질의를 여러 문서 데이터 베이스들에 동시에 브로드캐스트(broadcast) 하고 그 문서 데이터 베이스들에 의해 제공되는 결과들을 사용자에게 웹 문서 형태로 제시한다.

그러나 분산된 문서 데이터 베이스들은 많은 경우에 주제별로 구축되므로 웹 문서에 대한 색인들은 주제별로 군집화(clustering) 되어 주어진 질의에 대하여 사용자가 원하는 문서들을 제공하는 문서 데이터 베이스들의 개수는 하나 또는 몇 개에 지나지 않는 경우가 많다. 따라서 사용자 질의의 무분별한 브로드캐스트는 불필요한 네트웍 자원의 접근과 함께 상당한 통신 비용을 발생시킨다. 그러므로 효율적인 분산 정보 검색을 위해서는 사용자가 원하는 문서를 제공할 것으로 판단되는 문서 데이터 베이스들에게 선택적으로 질의를 주어 그 문서 데이터 베이스들로부터 문서를 검색하는 방법이 필요하다.

위와 같은 웹 정보 검색 에이전트들의 고찰을 통하여 분산 정보 검색 영역에서 데이터 베이스 선택 문제(database selection problem) 라고 불리는 흥미로운 문제를 도출해 낼 수 있다. 이것은 분산된 여러 개의 문서 데이터 베이스들 중에서 사용자가 원하는 문서를 제공하는 문서 데이터 베이스를 어떻게 골라낼 것인가?" 하는 문제로 표현된다. 이러한 데이터 베이스 선택 문제를 해결하기 위하여 여러 방법들이 제시되었다.

SMART[7] 시스템에서는 각 문서 데이터 데이터 베이스의 문서들의 중심점 용어 벡터(centroid term-vector)[1]를 그 문서 데이터 베이스의 색인으로 사용하고 이에 대한 사용자 질의의 유사도를 측정하는 방법을 사용한다. 이러한 방법은 문서들이 주제에 따라 각 문서 데이터 베이스들에 잘 분류되어 있을 때 유용하다. 그러나 이러한 방법은 문서 데이터 베이스 내의 문서들에 대한 각 용어들의 분포 상황을 전혀 고려하고 있지 못하므로 실제 정보 검색 환경에서 잘못된 결과를 보여주는 경우가 많다.

SavvySearch[8]에서는 사용자 질의에 대한 정보 검색 결과들을 기록하고 이러한 경험적 기록을 훈련 데이터로 사용하여 각 문서 데이터 베이스와 질의 용어(query term)의 관련도를 증강 학습법(reinforcement learning method)으로 구하고 이를 바탕으로 임의의 질의에 대한 각 문서 데이터 베이스의 관련도를 계산하는 방법을 사용한다. 이러한 방법은 사용자로부터 정보 검색 지식을 학습하므로 사용자에 대해 적응성을 갖는다. 그러나 각 문서 데

이터 베이스에서 용어들의 상호 관련성(correlation)을 고려하지 못하고 있으므로 여러 개의 용어들로 구성된 질의에 대한 정보 검색에서는 효과적이지 못한 경우가 많다.

GlOSS[9] 와 gGlOSS[10] 에서는 각 용어에 대한 문서 빈도(document frequency)나 용어 가중치 합(sum of term weights)과 같은 통계 정보를 기반으로 하여 사용자 질의에 대한 각 문서 데이터 베이스의 관련도를 계산하는 방법을 사용한다. 이 방법은 실제 특정 환경에서 상당히 효과적임이 실험적으로 알려져 있으나 각 문서 데이터 베이스 내의 문서들에 대한 각 용어들의 분포 상황과 관련하여 매우 제한적인 가정들을 기반으로 하고 있다. 따라서 이러한 가정들을 기반으로 사용자 질의에 대하여 각 데이터 베이스를 확률적으로 평가하는 GlOSS 와 gGlOSS 의 방법들은 이러한 가정을 만족하지 않는 많은 정보 검색 환경에서는 효과적이지 못하다.

데이터 베이스 선택 문제를 해결하기 위한 위와 같은 기존의 여러 기법들의 한계를 극복하기 위해서 본 연구에서는 신경망 학습 메카니즘을 분산 정보 검색에 적용한 신경망 기반 웹 정보 검색 에이전트[11](이후로 "신경망 에이전트")를 개발하였다. 신경망 에이전트는 내부 지식 메카니즘으로 표준 신경망 메카니즘으로 불리는 역전파 신경망(BPN: BackPropagation neural Net)[12]을 적용하여, 사용자의 질의에 대한 정보 검색 결과와 그에 대한 피드백으로부터 정보 검색 지식을 학습하고, 학습된 지식을 바탕으로 새로운 질의에 대해 관련된 문서 데이터 베이스를 찾아내어 그로부터 정보를 검색한다.

본 논문에서는 이러한 신경망 에이전트의 사용자 피드백 학습 메카니즘을 소개하고 이의 유용성을 다른 기존의 분산 정보 검색 기법과의 비교 실험을 통하여 입증하고자 한다. 이를 위하여 2절에서는 관련 연구로서 기존 정보 검색 에이전트에서의 학습 기법 적용 사례들을 간단히 서술한다.

2. 정보 검색 에이전트에서의 학습

현재 널리 알려진 대부분의 정보 검색 에이전트는 학습 능력을 갖기 위해 귀납적 기계 학습(inductive machine learning)방식을 사용하고 있으며 최근에는 귀납적 기계 학습에서 이용되고 있는 엔트로피 (entropy)개념을 응용한 다양한 방식들이 연구되고 있다. 정보 검색 에이전트는 최소한의 정보 검색 환경에 대한 배경 지식을 바탕으로 사용자의 행위 관찰, 피드백, 훈련, 그리고 다른 에이전트의 충고 등을 통한 학습 방법들을 통하여 효율적인 정보 검색에 필요한 지식을 습득한다[13].

사용자의 관찰을 통한 학습 방법은 에이전트가 사용자의 행위를 계속적으로 관찰하여 필요한 지식을 습득하고 학습하는 방식이다. 이때 정보 검색 에이전트는 오랜 기간 동안 사용자의 행위를 감시하고, 반복되는 행위의 패턴을 자동화한다. 예를 들어서 사용자가 A로부터 오는 대부분의 메일은 일고, B로부터 오는 대부분의 메일은 읽지 않고 삭제한다면 에이전트는 어떠한 메일을 어떻게 처리해야 하는지에 대한 지식을 습득할 수 있다. 비슷한 예로 뉴스 필터링 에이전트(news filtering agent)는 사용자가 주로 읽는 기사의 패턴을 학습하여 그와 비슷한 기사가 발견되면 사용자에게 그 기사를 제공할 수 있다. 이를 위한 대표적인 학습 기법은 기억 기반 학습(memory-based learning)이다. 사용자가 특정한 행동을 수행할 때 에이전트는 사용자가 수행한 모든 상황-행위(situation-action)들을 메모리에 저장한다. 새로운 상황이 발생했을 때 에이전트는 그 상황을 저장된 상황-행위와 비교함으로써 가장 차이가 가장 작은 행위를 취하게 된다.

사용자의 피드백을 통한 학습은 사용자의 직접적 또는 간접적인 피드백을 이용하는 학습 방법으로 정보 검색 에이전트의 적응성을 위해 자주 사용되어지는 방법이다[14].

사례를 통한 학습 방식은 사용자가 의도적으로 사례를 제시하여 정보 검색 에이전트를 학습시키는 것이다. 사용자는 정보 검색 에이전트에게 가상의 사건, 상황에 대한 사례를 제시하고 그러한 상황에서는 무엇을 해야 할 것인지를 보이면서 정보 검색 에이전트를 훈련시키는 것이다. 정보 검색 에이전트는 새로운 사례와 기존 사례 사이의 상관관계를 계산하고 사례 베이스를 적절히 변화시키면서 새로운 사례를 수용한다.

정보 검색 에이전트가 필요한 지식을 얻는 또 하나의 방법은 비슷한 일을 수행하는 다른 에이전트에게 충고를 요청하는 것이다. 한 에이전트가 어떤 특수한 상황에서 무엇을 해야 할지 알지 못한다면 다른 에이전트에게 그 상황을 제시하여 충고를 요청할 수 있다. 따라서 이러한 방법은 상호 협동적인 다중 에이전트들의 형태로 구현된다[15].

위와 같은 여러 가지 학습 방법들을 적용하여 다수의 웹 정보 검색 에이전트가 개발되었다.

WebWatcher 는 웹에서 사용자가 원하는 정보를 검색할 수 있도록 도와주는 지능형 에이전트로 카네기 멜론 대학에서 만들었다. 정보를 검색할 때, 각 웹 페이지내의 링크를 따라 정보를 검색하는데, 사용자가 현재 보고 있는 페이지내의 링크들 중 가장 유력한 링크를 다음 검색 페이지로 추천한다. 이때 사용자는 추천하는 링크를 선택할 수도 있고, 임의의 다른 링크를 선택할 수도 있다. WebWatcher 는 추천 링크에 대한 사용자의 반응과 웹 링크를 통해 원하는 정보를 획득했는 지의 성공 여부를 학습 예제 집합으로 기록한다. 이렇게 형성된 학습 예제 집합들에 기계학습을 적용하여 WebWatcher 는 검색 제어 지식을 자동적으로 학습한다. 시간이 지남에 따라, 경험에 의한 더 많은 검색 제어 지식을 학습하게 되어, 그 주제 분야의 웹 검색에는 전문가가 되는 것이다. 그러므로 WebWatcher 는 학습 도제(learning apprentice) 시스템의 일종이다[16].

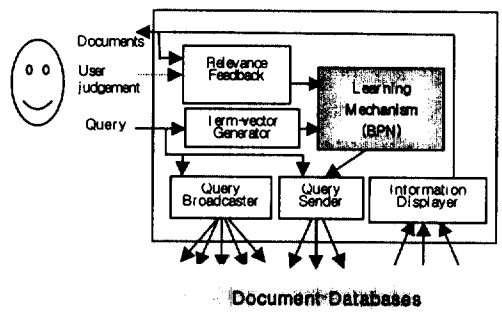
Syskill&Webert 는 사용자의 관심에 대한 취향을 사용자 행위의 관찰과 피드백을 통하여 학습하고 사용자가 관심을 가지는 웹 페이지를 구별해 내는 적응형 소프트웨어 에이전트이다[17]. 이 에이전트는 해당 웹 페이지와 사용자의 관심도와의 연관성 등급 입력, 인덱스 페이지로의 복귀 또는 주제 재 설정, 현재 주제에 대한 명시적인 사용자 취향의 학습, 제안의 생성 또는 LYCOS 검색 엔진 질의 등의 서비스를 제공한다. 학습 알고리즘으로는 주어진 개념에 대한 양의 예제(positive example)의 집합과 음의 예제(negative example)의 집합을 사용하고 웹 페이지들을 이진 벡터로 나타내는 부울 모델(boolean model)[18]을 사용하였다. 한편 이와 비슷한 기계 학습 방식을 사용하는 벡터 공간 모델 기반 적응형 웹 브라우징 에이전트도 소개되었다[19].

이외에도 또한 사용자로부터 취향을 학습하여 개인 신문과 같은 사용자의 취향에 맞는 문서들을 자동적으로 제공해 주거나 사용자가 흥미를 가지는 문서들만을 검색하도록 도와주는 적응형 프록시(proxy)에이전트로서 WebMate가 개발되었다[20].

또한 동적인 환경에서의 웹 정보 검색 에이전트와 같은 적응형 에이전트를 구현할 수 있는 프레임워크에 관한 연구들도 활발히 이루어지고 있는데 SEPIA[21]와 Soar[22]등이 그것들이다.

3. 신경망 에이전트

신경망 에이전트는 질의를 받고 자신의 색인 정보에 기반하여 그 질의에 관련하여 문서를 제시하는 다수의 문서 데이터 베이스들이 존재하는 환경에서 동작한다. 따라서 신경망 에이전트는 존재하는 문서 데이터 베이스들에 선택적으로 질의를 보내고 그들로부터 질의에 관련된 문서들을 제공받는다. (그림 1)은 신경망 에이전트의 주요 모듈들과 그들 사이의 제어 흐름을 보여준다. 따라서 하나의 신경망 에이전트는 6-tuple, = <QB, ID, RF, TG, LM, QS>로 정의되며 각 모듈들은 아래와 같이 나타내어 진다.



(그림 1) 신경망 에이전트의 구조

질의 브로드캐스터(QB: Query Broadcaster) : QB는 질의 제공자로부터 주어진 질의에 대하여 관련된 문서들을 제공받기 위해 존재하는 모든 문서 데이터 베이스들에 주어진 질의를 브로드캐스트한다.

정보 디스플레이어(ID: Information Displayer) : ID 는 문서 데이터 베이스들로부터 반환되어지는 문서들의 중복성을 제거하여 합병한 결과 (반환되어진 모든 문서들의 합집합) 를 사용자에게 질의 제공자

에게 디스플레이 한다.

관련도 피드백(RF: Relevance Feedback) : RF 는 ID 에 의해 디스플레이된 문서들의 주어진 질의 q 에 대한 관련도를 평가하여 벡터 $C_q=(c_{q1}, c_{q2}, \cdots, c_{qM})$ 를 생성시킨다. 이 때, D를 존재하는 모든 문서 데이터 베이스들의 순서 집합, $D=\{d_1, d_2, \cdots, d_M\}$, 이라고 하고 m_{qi}를 질의 q에 대하여 문서 데이터 베이스 d_i가 제공하는 관련된 문서들의 개수라고 한다면,

$$\text{for } i=1,\cdots,M,\ c_{qi}=\begin{cases}\dfrac{m_{qi}}{\max_{j=1,\cdots,M} m_{qj}} & \text{if } \max_{j=1,\cdots,M} m_{qj}\neq 0\\ 0 & \text{otherwise}\end{cases}$$

이 성립한다.

m_{qi}는 질의 q에 대하여 문서 데이터 베이스 d_i가 반환하는 문서들의 개수를 구함으로써 쉽게 구할 수 있다. 그러나 문서 데이터 베이스의 검색 결과로서 반환되는 모든 문서들이 항상 주어진 질의와 관련되는 것은 아니다. 일반적으로 문서가 주어진 질의에 대하여 관련된다는 것은 상당히 주관적이어서 그 질의 제공자에 의해서만 정확히 판단될 수 있다. 다시 말하면 주어진 질의에 대해 문서 데이터 베이스가 반환하는 문서들이 모두 질의 제공자가 원하는 것은 아니라는 것이다. 본 신경망 에이전트는 여러 문서 데이터 베이스들로부터 질의 제공자가 원하는 문서들만을 효율적으로 검색하는 것을 궁극적인 목표로 하고 있다. 그러므로 m_{qi}를 q에 대하여 d_i가 제공하는 문서들 중 질의 제공자에 의해 q에 관련된다고 판단되는 문서들의 개수로 한다면 신경망 에이전트의 정보 검색 성능을 보다 향상시킬 수 있을 것이다. 이것은 실제 웹 정보 검색 환경에서 질의 q에 대한 각 문서 데이터 베이스의 검색 결과에 대하여 질의 제공자로부터 관련도 피드백을 얻음으로써 구할 수 있다.

용어 벡터 생성기(TG: Term-vector Generator) : 용어 벡터 생성기는 불용어를 제거하고 복수 명사를 단수로 바꾸거나 변형된 동사를 원형으로 바꾸는 등의 스테밍(stemming) 과정[1]을 통해, 용어 집합으로 표현되는 질의 q를 이진 벡터 $S_q=(s_{q1}, s_{q2}, \cdots, s_{qN})$로 변환한다. 이 때, T를 용어 집합 $T=\{t_1, t_2, \cdots, t_N\}$ 이라 한다면,

$$\text{for } i=1,2,\cdots,N,\ s_{qi}=\begin{cases}1 & \text{if } t_i \text{ occurs in } q\\ 0 & \text{otherwise}\end{cases}$$

이 성립한다.

학습 메카니즘(LM: Learning Mechanism) : 신경망 에이전트는 검색 결과에 대한 피드백으로부터 분산 정보 검색 지식을 학습하고 새로운 검색을 수행할 때 학습된 검색 지식을 기억해내기 위해 신경망 연상 메모리 (neural network associative memory)의 형태로 학습 메카니즘을 가진다. 이를 위해 안정된 학습기억 능력을 가지고 있는 표준 신경망으로 평가되는 역전파 신경망 (BPN: BackPropagation Neural net) 을 사용한다. 이러한 학습 메카니즘 모듈은 분산 정보 검색을 하기 위한 사용자 피드백 학습 메카니즘의 중심이 되며 이에 대한 자세한 설명은 4 절에서 언급한다.

질의 발송기 (QS: Query Sender) : QS 는 주어진 질의를 BPN 검색 단계의 출력에 기반하여 이용 가능한 문서 데이터 베이스들에게 다음과 같이 선택적으로 보낸다.

D 를 존재하는 모든 문서 데이터 베이스들의 순서 집합, $D=\{d_1, d_2, \cdots, d_M\}$, 이라 하고 O_q 를 주어진 질의 q 에 대한 BPN 검색 단계의 출력 벡터, $O_q=(o_{q1}, o_{q2}, \cdots, o_{qM})$, 라고 하며 τ 를 0 과 1 사이의 허용 오차 상수라고 한다면, i = 1, 2, ... , M 에 대하여 $o_{qi} \geq \tau$ 인 경우에만 QS 는 q 를 d_i 에 보낸다.

4. 사용자 피드백 학습 메카니즘

신경망 에이전트의 사용자 피드백 학습 메카니즘은 (그림 2)과 같이 훈련 단계 (training phase) 와 검색 단계 (retrieval phase) 의 두 가지 단계로 동작한다. 이 그림에서 진하게 표현된 부분은 신경망 에이전트내부의 BPN 학습 메카니즘 모듈을 나타낸다.

4.1 훈련 단계

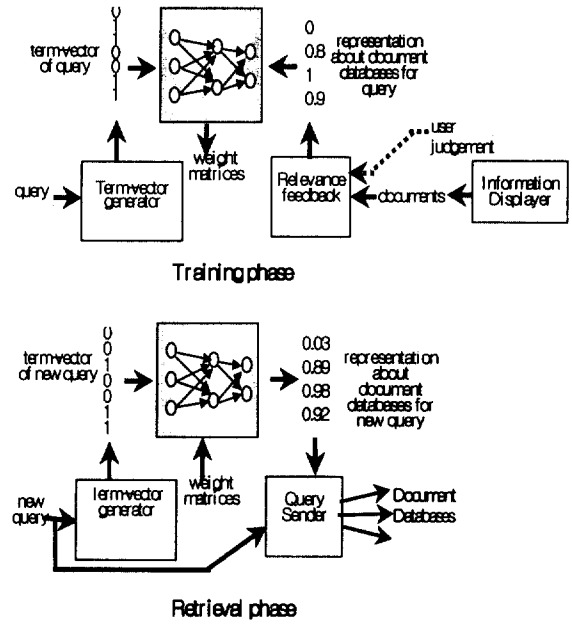
훈련 단계에서 BPN 의 입력층은 TG 에 의해 생성되어진 S_q 를 나타내고, 출력층은 RF 에 의해 생성되어진 C_q 를 나타내며 이들은 하나의 훈련 쌍 (S_q, C_q) 을 이룬다. 주어진 훈련 질의들에 대한 TG 와 RF 의 출력들로 이루어진 모든 훈련 쌍들에 대해 역전파 알고리즘[23]을 적용한다. 신경망 에이전트에서 역전파 알고리즘은 BPN 내부 노드들간의 연결 가중치들을 조정함으로써 각 문서 데이터 베이스가 제공하는 문서들의 관련도에 대해 학습한다. 이러한 사용자 피드백 학습 메카니즘의 훈련 단계를 수행하기 위한 신경망 에이전트 α의 전체적인 훈련 절차를 (그림 3)에서 나타낸다.

QB 는 주어진 질의를 모든 이용 가능한 문서 데이터 베이스들에게 브로드캐스트하고 문서가 반환되기를 기다린다. ID는 반환되는 모든 문서들을 합병하고 질의 제공자에게 디스플레이한다. RF는 ID에 의해 디스플레이되는 문서들로부터 주어진 질의에 관련된 문서들을 검사하고 각 문서가 어느 문서 데이터 베이스로부터 왔는지를 알아내어 결과적으로 각 문서 데이터 베이스가 주어진 질의에 관련된 문서들을 얼마나 많이 제공하였는지를 표현하는 벡터를 생성한다. 주어진 질의에 대하여 RF에 의해 생성되는 벡터는 TG에 의해 생성되는 용어 벡터와 함께 하나의 훈련 쌍을 형성한다.

이와 같은 방법으로 질의 제공자에 의해 주어지는 모든 훈련 질의들로부터 생성되는 훈련 쌍들의 집합으로 신경망 에이전트 내부의 BPN 을 훈련시킨다. 이러한 과정을 통하여 신경망 에이전트는 문서 데이터 베이스들로부터 반환되는 문서들에 대하여 관련도 피드백을 받아 주어진 질의에 관련되는 문서가 어디에 있는지에 대한 지식을 습득한다.

4.2 검색 단계

검색 단계에서는 주어진 질의에 대하여 TG 에 의해 생성되어진 용어 벡터가 BPN 의 입력층에 주어지고 이는 훈련 단계에서 학습되어진 노드들간의 연결 가중치를 기반으로 하여 은닉층을 통하여 출력층으로 전파된다. 이의 결과로 BPN은 0과 1사이의 값들로 이루어지는 벡터를 출력한다. 이러한 사용자 피드백 학습 메카니즘의 검색 단계를 수행하기 위한 신경망 에이전트 α 의 전체적인 검색 절차를 다음과 같이 나타낸다.



(그림 2) 사용자 피드백 학습 메카니즘의 훈련 단계와 검색 단계

```
Procedure ATrains( α ) :
Let α = < QB,ID,RF,TG,LM,QS > // LM is BPN
begin
trainingpairset ← ( )
for each query q given by the query issuer
  begin
    TG generates the term-vector Sq ;
    QB broadcasts q to available document databases ;
    Wait for all documents submitted by the document
      databases ;
    ID displays the union of all the documents submitted by
      the document databases to the query issuer ;
    RF produces the vector representation Cq from relevance
      feedback ;
    trainingpairset ← trainingpairset ∪ { (Sq , Cq) ) ;
  end
    Train LM with trainingpairset by the backpropagation
      algorithm ;
end
```

(그림 3) 신경망 에이전트의 훈련 절차

신경망 에이전트 α = < QB, IM, RF, TG, LM, QS > 는 질의 제공자로부터 용어들로 표현되는 질의를 받아서 다음과 같은 단계에 따라 그 질의에 대해 문서들을 반환한다.

단계 1: TG 는 주어진 질의 q 를 용어 벡터 S_q

로 반환한다.

단계 2: S_q 에 의해 활성화된 LM (BPN) 는 검색 단계에 의해 O_q 를 출력한다.

단계 3: QS는 O_q를 기반으로 하여 문서 데이터 베이스들을 선택하고 선택된 문서 데이터 베이스들에 q를 보낸다.

단계 4: ID는 단계 3에서 선택된 문서 데이터 베이스들로부터 반환되는 모든 문서들을 합병하고 그 결과를 질의 제공자에게 제시한다.

단계 2와 3에서 신경망 에이전트는 BPN의 연결 가중치 행렬로서 저장된 지식을 이용하여 주어진 질의에 관련된 문서들을 반환할 것으로 판단되는 문서 데이터 베이스들을 찾아내고 그 문서 데이터 베이스에 질의를 보내어 정보 검색을 수행한다.

5. 실험

5.1 실험 방법

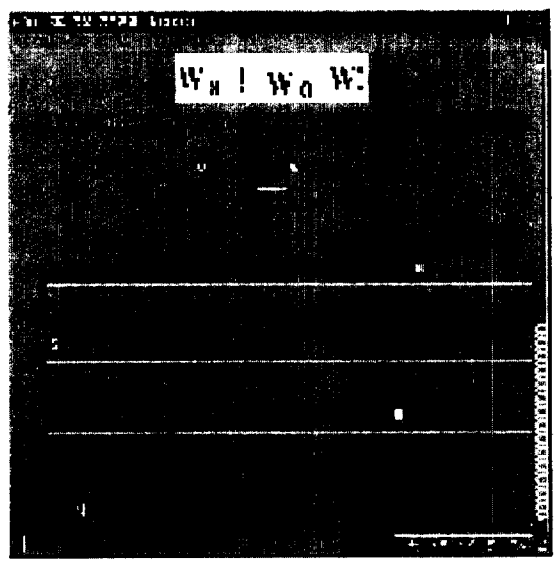
야후! 코리아(Yahoo! Korea)는 다양한 카테고리들에 따라 계층적으로 구성된 검색 디렉토리들을 제공한다. 이러한 검색 디렉토리들 각각은 주어진 질의에 대해 특정한 카테고리에서 관련 문서들의 요약을 제공하므로 하나의 문서 데이터 베이스로서 동작한다. 그러므로 제안된 신경망 에이전트 기법의 성능을 실제 분산 정보 검색 환경에서 평가하기 위해 야후! 코리아의 검색 디렉토리들을 문서 데이터 베이스들로 사용하는 신경망 에이전트 기반 정보 검색 시스템을 구현하였다. <표 1>은 본 실험에서 사용된 16개의 문서 데이터 베이스 (야후! 코리아의 디렉토리) 들을 보여주고 있다.

<표 1>과 같은 16개의 문서 데이터 베이스 위에서 작동하는 신경망 에이전트 정보 검색 시스템 (SNA : Single Neural net Agent information retrieval system) 을 구축하였다. 웹을 통해 접근하여 사용할 수 있게 한 SNA 에 26일 동안 133명의 사용자가 용어들의 부울 논리곱(boolean AND) 형태의 797 개 질의를 주었다. 각각의 질의에 대하여 SNA는 훈련 단계에서 16개의 문서 데이터 베이스들을 무차별적 (exhaustive) 으로 검색하여 사용자의 관련도 피드백으로부터 몇 개의, 때로는 0 개의, 관련된 문서들을 확인하였다. 이 때, 사용자에게 신경망 에이전트의 ID 에 의해 제시된 문서들을 모두 검사하도록 요청하였다.

<표 1> 야후! 코리아의 16 개 문서 검색 디렉토리

디렉토리 카테고리	문서의 개수	디렉토리 카테고리	문서의 개수
물리학	102	경제학	129
화학	100	심리학	48
생물학	314	지리학	67
천문학	91	건축	89
전기 전자공학	217	공연예술	165
컴퓨터공학	195	스포츠	206
기계공학	114	전통예술	105
재료공학	56	의학	343

(그림 4)는 SNA 에서 질의 "지리 정보" 에 대해 반환된 문서 요약들에 대하여 사용자로부터 관련도 피드백을 받아들이는 사용자 인터페이스를 보여주고 있다.



(그림 4) 신경망 에이전트의 관련도 피드백 인터페이스

이러한 과정을 통하여 사용자가 실제로 신경망 에이전트의 ID에 의해 제시된 문서들을 모두 검사하여 적어도 하나 이상의 관련된 문서가 확인된 질의들만

을 실험에서 고려하였다. 결과적으로 질의와 그에 관련된 문서들로 이루어진 734개의 예제 쌍들을 수집하였다. 수집된 예제 쌍들에서 질의들의 토픽은 16개의 문서 데이터 베이스 카테고리들을 고르게 포함하였고 각 질의에 대하여 관련된 문서들은 하나의, 때때로 불과 몇 개의, 문서 데이터 베이스에 의해서 제공되었다. 734개의 예제 쌍들로부터 657개의 예제 쌍들을 임의로 골라내어 훈련 예제들로 사용하였고 나머지 77개의 예제 쌍들을 분산 정보 검색 성능 평가를 위한 테스트 예제들로 사용하였다. 훈련 예제들로부터 657개의 훈련 질의들과 그에 대한 사용자의 관련도 피드백 결과를 생성하고 이들을 사용하여 SNA를 훈련시키고 테스트 예제로부터의 77개의 테스트 질의들에 대하여 각각 성능을 평가하였다. 그리고 BPN 입력층의 크기는 734개의 질의들에 나타나는 모든 용어들의 개수, 221, 로 하였고 BPN 출력층의 크기는 문서 데이터 베이스들의 전체 개수, 16, 으로 하였다.

제시된 신경망 에이전트 기반 정보 검색 기법을 기존의 시스템들과 비교하기 위해서 SNA 에서 사용된 것과 같은 문서 데이터 베이스들과 테스트 예제들에 대하여 1 절에서 소개한 기존의 데이터 베이스 선택 기법, SMART, GlOSS, SavvySearch, 들의 성능도 실험을 통하여 평가하였다. 기존 시스템들의 데이터 베이스 선택 기법들은 각 질의에 대하여 0 보다 크거나 같은 수로 각 문서 데이터 베이스의 순위 값(rank) 을 생성한다. 이러한 기존 시스템들이 생성하는 순위 값들을 본 연구에서 제시한 방법과 서로 비교할 수 있게 하기 위해 의해 다음과 같이 0 과 1 사이의 값으로 정규화하였다.

D 는 모든 문서 데이터 베이스들의 순서 집합 $D=\{d_1, d_2, \cdots, d_M\}$ 이고 r_{qi}는 주어진 질의 q에 대하여 어떤 데이터 베이스 선택 기법에 의해 생성된 문서 데이터 베이스 d_i의 평가 값이라고 한다면, r_{qi} 는 다음과 같은 식에 의해 n_{qi} 로 정규화한다.

$$\text{for } i=1,\cdots,M,\ n_{qi}=\begin{cases}\dfrac{r_{qi}}{\max_{1\le j\le M} r_{qj}} & \text{if } \max_{1\le j\le M} r_{qj}\neq 0\\ 0 & \text{otherwise}\end{cases}$$

이렇게 정규화 되어진 순위 값 n_{qi} 을 실제 분산 정보 검색에 적용하기 위해 주어진 질의를 어떠한 0 과 1 사이의 허용 오차 상수 τ에 대하여 $n_{qi} \ge \tau$ 인 문서 데이터 베이스 d_i에만 질의를 보내는 전송 메카니즘으로서 3절에서 기술한 신경망 에이전트의 QS 를 사용하였다.

5.2 성능 평가

신경망 에이전트와 기존의 기법들을 사용하여 분산 정보 검색을 수행한 결과에 대한 성능을 평가하기 위해 정보 검색의 표준 성능 평가 척도로서 사용되는 정확률(precision) 과 재현률(recall) 을 다음과 같이 정의하였다.

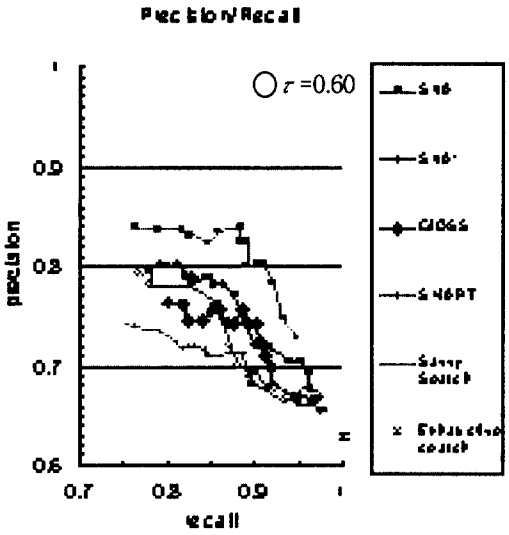
$$\text{정확률} = \frac{\text{주어진 질의에 관련된 검색된 문서들의 개수}}{\text{검색된 전체 문서들의 개수}}$$

$$\text{재현률} = \frac{\text{주어진 질의에 관련된 검색된 문서들의 개수}}{\text{테스트 예제에 존재하는 주어진 질의에 관련된 모든 문서들의 개수}}$$

각 기법들에서 QS의 허용 오차 상수 τ를 다양하게 변화시키면서 77개의 테스트 질의들 전체에 대하여 얻어진 평균 정확률과 재현률 평가 결과를 (그림 5)에서 나타내었다.

일반적으로 예상되는 바와 같이 QS의 허용 오차 상수 τ가 0.05(각 정보 검색 성능 곡선의 맨 오른쪽 점) 에서 0.05의 간격으로 0.95(각 정보 검색 성능 곡선의 맨 오른쪽 점)까지 증가함에 따라 정확률은 향상되고 재현률은 떨어진다. τ가 0인 경우는 모든 데이터 베이스 선택 기법들이 항상 사용자 질의를 모든 문서 데이터 베이스들에 무차별적으로 브로드캐스트하게 되며 이 때의 정보 검색 성능은 (그림 5)에서 무차별 검색(exhaustive search)으로 나타내고 있다. 또한 (그림 5)에서 SNA*로 나타내어진 결과는 SNA의 학습을 위한 피드백으로서 사용자로부터의 피드백을 사용하지 않고 훈련 질의에 대하여 단순히 각 문서 데이터 베이스가 반환하는 문서들의 개수[2]를 사용한 경우를 보여주고 있다. 이 경우는 각 문서 데이터 베이스의 검색 결과로서 반환되는 모든 문서들이 항상 주어진 질의와 관련된다는 다소 비현실적인 가정을 바탕으로 하고 있다.

2) 실제로는 각 문서 데이터 베이스가 반환하는 문서들의 개수를 0 과 1 사이로 정규화한 값을 사용한다.



(그림 5) 정확률과 재현률 평가 결과

(그림 5)으로부터 사용자 피드백 학습 메카니즘을 적용한 신경망 에이전트 기법, SNA, 이 기존의 기법들, SMART, GlOSS, SavvySearch, 에 비해 정보 검색 성능이 뛰어나며 사용자로부터의 피드백을 사용하지 않은 신경망 에이전트 기법, SNA*, 은 기존의 기법에 비해 더 나은 성능을 보이지 않는다는 것을 알 수 있다. 한편 GlOSS는 SMART에 비해 τ가 크지 않을 때 더 좋은 정보 검색 성능을 보이고 τ가 클 때는 다소 나쁜 정보 검색 성능을 보인다. 그리고 SavvySearch의 정보 검색 성능은 다른 기법들에 비해 전반적으로 좋지 않음을 알 수 있다. 이것은 SavvySearch가 사용하는 증강 학습법이 정보 검색 영역에서 일반적으로 상호 간섭 현상(cross-talk effect) 과 같은 문제점을 가진다는 사실로 설명될 수 있다.

τ가 0.60 일 때 신경망 에이전트 기법은 무차별 검색에 비해 정확률에 있어서 32% 이상의 향상을 보여주고 재현률에 있어서는 12% 이하의 저하를 보여준다. 많은 경우에 있어서 정보 검색 작업은 "too-much problem" 이라고 불리는 문제로 어려움을 겪는다. 이 문제는 주어진 질의에 대하여 정보 검색 시스템은 정상적으로 처리하기가 힘들 정도로 너무 많은 문서들을 반환하는 것과 관련된 문제이다 [24]. 이것은 많은 정보 검색 환경에서 정확률이 재현률보다 더 중요한 성능 평가 척도임을 의미한다. 그러므로 재현률의 저하보다 더 큰 정확률의 향상은 정보 검색 성능에 있어서 중대한 의미를 가진다.

SNA의 BPN 훈련을 위해 걸린 시간은 Solaris 2.7에 의해 운영되는 Sun Ultra 60 Workstation에서 5분을 넘지 않았다. 일반적인 정보 검색 시스템의 정보 검색 지식 학습 또는 색인 과정은 일괄 처리 / 오프 라인(batch / off-line) 방식에 의해 주기적으로 수행되므로 수행 시간이 결정적으로 중대한 요소가 아니라는 것을 고려한다면 실제로 정보 검색 지식을 학습하는데 걸린 이와 같은 시간은 충분히 받아 들일만 하다.

6. 결론 및 향후 연구

본 논문에서는 분산 정보 검색의 테이터 베이스 선택 문제 해결을 위한 신경망 기반 사용자 피드백 메카니즘을 제시하였다. 제시된 메카니즘은 분산 정보 검색 지식을 습득하기 위해 역전파 학습 알고리즘을 이용하여 사용자 질의에 관련된 문서들을 제공할 문서 데이터 베이스를 찾아낸다.

실험으로부터 본 논문에서 제시된 사용자 피드백 메카니즘을 적용한 신경망 에이전트 시스템이 기존의 여러 기법들에 비해 현저하게 성능을 향상시킬 수 있음을 확인하였다.

본 논문에서 제시한 신경망 에이전트 기법은 기존의 통계적 기법들에 비해 다음과 같이 두 가지 근본적인 장점을 가진다.

○ 질의 공간을 추상화하기 위하여 사용자의 관련도 피드백으로부터 정보 검색 지식을 습득하는 신경망 에이전트는 주어진 질의에 대하여 사용자의 흥미와 관련된 문서들을 제공하는 문서 데이터 베이스를 찾을 수 있다. 이 때, 사용자는 단일 사용자이거나 공통된 흥미를 가지는 사용자 그룹을 뜻하며 신경망 에이전트는 사용자에게 종속적인 개인용 정보 검색 에이전트로서 동작한다. 본질적으로 사용자의 흥미는 매우 주관적이므로 주어진 질의에 대하여 문서들이 사용자의 흥미에 관련되는 지에 대한 정확한 결

정은 사용자에 의해서만 행해질 수 있다. 따라서 사용자 피드백을 사용하지 않는 기법들은 사용자로부터의 관련도 피드백을 이용하지 않으므로 사용자의 흥미와 관련된 문서들을 제공하는 문서 데이터 베이스를 찾는데 항상 효과적이지는 않다.

○ 신경망 에이전트 기법은 자신의 색인 정보를 외부 시스템에 제공하는 것을 허락하지 않는 비협동적 문서 데이터 베이스들에 대해서도 적용될 수 있다. 이것은 신경망 에이전트가 문서 데이터 베이스들에 의해 반환되는 검색 결과들에 대한 관련도 피드백으로부터 학습을 수행하므로 문서 데이터 베이스들로부터 어떠한 색인 정보(또는 색인 정보에 대한 통계 값)도 필요로 하지 않기 때문에 가능하다.

본 논문에서 제시된 신경망 기반 학습 메카니즘을 동적인 특성을 갖는 정보 검색 영역에도 적용할 수 있게 하기 위해 웹과 같이 동적이면서 분산된 문서 데이터 베이스들로부터 정보 검색 지식을 자동으로 추출하는 웹 로봇에 관한 연구가 준비중이다. 또한 이미 존재하는 다양한 형태의 정보 자원을 문서 데이터 베이스로서 학습할 수 있게 하기 위해 랩핑(wrapping) 기술에 대한 연구를 병행할 계획이다. 랩핑 기술을 적용하여 구현되는 랩퍼(wrapper) 는 검색 질의들을 기존의 정보 자원이 이해할 수 있는 질의나 명령으로 번역하고 정보 자원으로부터의 결과를 신경망 기반 학습 메카니즘에서 이용할 수 있는 형태로 변환하는 기능을 수행하게 될 것이다. 이러한 랩퍼를 이용함으로써 신경망 기반 학습 메카니즘은 단순한 문서 모임 뿐만 아니라 관계형 데이터 베이스와 같은 정보 자원들도 분산 정보 검색을 위해 활용할 수 있게 될 것이다.

참 고 문 헌

[1] G. Salton and M. McGill, Introduction to Modern Information Retrieval, MacGraw-Hill, New York NY, 1983.

[2] M. Mauldin and J. Leavitt, Web agent related research at the Center for Machine Translation, in Proceedings of ACM SIGNIDR '94, 1994.

[3] B. Pinkerton, Finding what people want: Experiences the webcrawler, in Proceedings of 2nd WWW Conference, 1994.

[4] Networked Computer Science Technical Reports Library, http://lite.ncstrl.org :3803/.

[5] IBM InfoMarket, http://www.infomarket.ibm.com/.

[6] E. Selberg and O. Etzioni, Multi-Service Search and Comparison Using the MetaCrawler, in Proceedings of 4th International WWW Conference, December 1995.

[7] G. Salton, The SMART Retrieval System - Experiments in Automatic Document Processing, Prentice-Hall, Inc., Englewood Cliffs NJ, 1971.

[8] A. Howe and D. Dreilinger, SavvySearch: A Meta-Search Engine that Learns Which Search Engines to Query, AI Magazine, 18(2), 1997.

[9] L. Gravano, H. Garcia-Molina, and A. Tomasic, The Effectiveness of GlOSS for the Text-Database Discovery Problem, in Proceedings of ACM SIGMOD, 1994.

[10] L. Gravano and H. Garcia-Molina, Generalizing GlOSS to Vector-Space Databases and Broker Hierarchies, in Proceedings of VLDB, 1995.

[11] Yong S. Choi and Suk I. Yoo, Multi-agent Learning Approach to WWW Information Retrieval using Neural Network, in Proceedings of ACM International Conference on Intelligent User Interfaces, 1999.

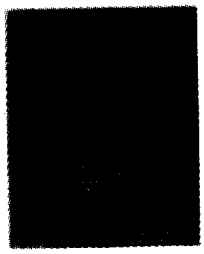
[12] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. thesis, Harvard University, Cambridge, MA, 1974.

[13] P. Maes, Agent that reduce work and information overhead, Comm. of the ACM, 37(7), 1994.

[14] B. Yuwono et. al., Server Ranking for Distributed Test Resource Systems on the Internet, in Proceedings of 5th International Conference on Database Systems For Advanced Applications (DASFAA'97),

Melbourne, April 1997.
[15] D. A. Maltz, Distributing information for collaborative filtering on Usenet net news, SM thesis, MIT, Cambridge, MA, 1994.
[16] R. Armstrong et. al., WebWatcher: Machine Learning and Hypertext, Fachgruppentreffen Maschinelles Lernen, Dortmund, Germany, 1995.
[17] M. Pazzani et. al., Syskill&Webert: Identifying Interesting Web Sites, in Proceedings of AAAI96, Portland, 1996.
[18] V.N. Gudivada, V.V. Raghavan, W.I. Grosky, and R. Kasanagottu, Information Retrieval on the World Wide Web, IEEE Internet Computing, 1997.
[19] Marko Balabanovic and Yoav Shoham, Learning Information Retrieval Agents: Experiments with Automated Web Browsing, in Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Resources, March 1995.
[20] Liren Chen and Katia Sycara, WebMate: A Personal Agent for Browsing and Searching, in Proceedings of the ACM International Conference on Autonomous Agents, 1998.
[21] Alberto Segre et. al., Planning, Acting, and Learning in Dynamic Domain, Machine Learning Methods for Planning, Morgan Kaufmann Publishers, 1993.
[22] John Laird et. al., Integrating Execution, Planning, and Learning in Soar for External Environments, in Proceedings of the AAAI90, 1990.
[23] J. A. Freeman and D. M. Skapura, Neural Networks Algorithms, Applications, and programming Techniques, Addison-Wesley, MA, 1992.
[24] B.A. LaMacchia, Internet Fish, PhD thesis, MIT, MA, 1996.

최 용 석



1993 서울대학교 전산과학과 (이학사)
1995 서울대학교 전산과학과 (이학석사)
2000 서울대학교 전산과학과 (이학박사)
2000.3~8 삼성전자 통신연구소 IMT-2000 연구개발팀
2000.9~현재 한양대학교 컴퓨터교육과 전임교수
관심분야: 지능형 정보검색, 소프트웨어 에이전트, 기계학습, 컴퓨터교육