분산 정보 검색을 위한 신경망 에이전트
(Neural Net Agent for Distributed Information Retrieval)

최 용 석 †
(Yong S. Choi)

요 약 웹과 같은 분산 정보 검색 환경에서 문서들은 많은 문서 데이타베이스들에 자연스럽게 분할되어서 존재한다. 그러므로 이러한 문서들의 효율적인 검색을 위해서는 먼저 질의에 관련되는 문서들을 제공할 것으로 판단되는 문서 데이타베이스를 찾아내고 다음으로 그 문서 데이타베이스에 질의를 줌으로써 분산 정보 검색을 수행해야 한다.

본 논문에서는 이러한 효율적인 분산 정보 검색을 위한 신경망 에이전트를 제안한다. 신경망 에이전트는 질의 검색 예제들을 통하여 얻어진 질의에 대한 관련도 피드백 정보에 기반하여 역전파 알고리즘으로 분산 정보 검색 지식을 학습한다. 충분히 학습한 후의 신경망 에이전트는 주어진 질의에 대하여 관련 문서 데이타베이스들을 찾아내고 그 문서 데이타베이스들로부터 관련되는 문서들을 검색한다.

실험에서는 제안된 신경망 에이전트 시스템을 구현하여 정보 검색 성능을 널리 알려진 기존의 분산 정보 검색 기법을 사용했을 때와 비교함으로써 신경망 에이전트의 유용성을 예증한다.

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 agent approach to such an efficient information retrieval. First, we present a neural net agent that learns about underlying document databases using the relevance feedbacks obtained from many retrieval experiences. For a given query, the neural net agent, which is sufficiently trained on the basis of the BPN learning mechanism, discovers the document databases associated with the relevant documents and retrieves those documents effectively.

In the experiment, we introduce a neural net agent based information retrieval system and evaluate its performance by comparing experimental results to those of the conventional well-known approaches.

1. 서 론

웹 상에 존재하는 엄청난 양의 문서 정보는 여러 문서 데이타베이스들에 분산된 형태로 존재한다. 이와 같이 문서 정보들이 여러 문서 데이타베이스들에 분산된 환경에서의 정보 검색을 분산 정보 검색이라고 한다. 이러한 분산 정보 검색에서 사용자는 원하는 문서를 찾기 위하여 알고 있는 모든 문서 데이타베이스들에 일일이 질의를 주어 검색해야 할 지도 모른다. 그러나 분산된 문서 데이타베이스들은 많은 경우에 주제별로 구축되어 문서 정보가 각 문서 데이타베이스에 주제별로 군집화되므로 주어진 질의에 대하여 사용자가 원하는 문서들을 제공하는 문서 데이타베이스들의 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 데이타베이스들에게만 선택적으로 질의를 주어 그 문서 데이타베이스들로부터 문서를 검색하는 방법이 필요하다.

위와 같은 고찰을 통하여 분산 정보 검색 영역에서 데이타베이스 선택 문제(database selection problem)를 도출해 낼 수 있다. 이것은 분산된 여러 개의 문서 데이타베이스들 중에서 사용자가 원하는 문서를 제공하는 문서 데이타베이스를 어떻게 골라낼 것인가?" 하는 문

† 정 회 원 : 한양대학교 컴퓨터교육과 교수
cys@hanyang.ac.kr
논문접수 : 2001년 2월 24일
심사완료 : 2001년 8월 2일

제로 표현된다. 이러한 데이타베이스 선택 문제를 해결하기 위하여 여러 방법들이 제시되었다.

가장 간단한 방법은 사용자가 직접 각 문서 데이타베이스를 브라우징 (browsing) 하게 하는 것이다. 널리 알려진 Prospero file system[1]과 Gopher[2] 등이 이러한 방법을 사용하고 있다. 이러한 방법은 사용자가 원하는 문서를 정확히 검색할 수 있다는 측면에서는 효과적일 수 있으나 여러 개의 문서 데이타베이스들에 존재하는 모든 문서들을 사용자가 일일이 살펴보아야 하므로 대규모 문서 모임을 관리하는 문서 데이타베이스 환경에서는 매우 비현실적이다.

또 다른 방법은 사용자가 각 문서 데이타베이스에 대한 메타 정보(meta information)를 가지고 있는 메타 데이타베이스에 질의를 하게 하는 것이다. 예를 들어 WAIS[3]는 각 문서 데이타베이스를 요약하는 메타 데이타베이스로서 directory of servers를 사용하며, World-Wide Web Worm[1]은 각 문서 데이타베이스에 존재하는 문서들의 제목 또는 본문의 일부를 메타 색인 정보로 이용한다. ALIWEB[2]시스템은 사용자가 생성한 각 문서 데이타베이스의 전체 내용에 대한 간략한 요약을 유지하는 방법을 사용한다. 이러한 방법의 유용성은 각 문서 데이타베이스에 대하여 메타 정보를 얼마나 효과적으로 구축하느냐에 달려있다. 그러나 효과적인 메타 정보는 문서 데이타베이스의 모든 내용을 간략하면서도 완전하게 요약하는 것이어야 하므로 구축하는데 현실적으로 상당한 어려움이 따른다.

SMART[4] 시스템에서는 각 문서 데이타 데이타베이스의 문서들의 중심점 용어 벡터(centroid term-vector)[5]를 그 문서 데이타베이스의 색인으로 사용하고 이에 대한 사용자 질의의 유사도를 측정하는 방법을 사용한다. 이러한 방법은 문서들이 주제에 따라 각 문서 데이타베이스들에 잘 분류되어 있을 때 유용하다. 그러나 이러한 방법은 문서 데이타베이스 내의 문서들에 대한 각 용어들의 분포 상황을 전혀 고려하고 있지 못하므로 실제 정보 검색 환경에서 잘못된 결과를 보여주는 경우가 많다.

SavvySearch[6]에서는 사용자 질의에 대한 정보 검색 결과들을 기록하고 이러한 경험적 기록을 훈련 데이타로 사용하여 각 문서 데이타베이스와 질의 용어(query term)의 관련도를 감독 증강 학습법(supervised reinforcement learning method)으로 구하고 이를 바탕으로 임의의 질의에 대한 각 문서 데이타베이스의 관련도를 계산하는 방법을 사용한다. 이러한 방법은 사용자로부터 정보 검색 지식을 학습하므로 사용자에 대해 적응성을 갖는다. 그러나 각 문서 데이타베이스에서 용어들의 상호 관련성(correlation)을 고려하지 못하고 있으므로 여러 개의 용어들로 구성된 질의에 대한 정보 검색에서는 효과적이지 못한 경우가 많다.

GlOSS[7]와 *gGlOSS*[8]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 데이타베이스의 관련도를 계산하는 방법을 사용한다. 이 방법은 실제 특정 환경에서 상당히 효과적임이 실험적으로 알려져 있으나 각 문서 데이타베이스 내의 문서들에 대한 각 용어들의 분포 상황과 관련하여 매우 제한적인 가정(assumption)들을 기반으로 하고 있다. 따라서 이러한 가정들을 기반으로 사용자 질의에 대하여 각 데이타베이스를 확률적으로 평가하는 *GlOSS*와 *gGlOSS*의 방법들은 이러한 가정을 만족하지 않는 많은 정보 검색 환경에서는 효과적이지 못하다.

본 논문에서는 이러한 기존 기법들이 데이타베이스 선택 문제의 해결을 위해 불충분함을 보이고 본 연구실에서 개발한 신경망 기반 웹 정보 검색 에이전트(이후로 신경망 에이전트)[9][10]가 보다 효과적임을 보이고자 한다.

이를 위하여 먼저 데이타베이스 선택 문제 해결을 위해 널리 사용되는 몇 가지 기법들을 보다 상세히 기술하고 한계점을 분석한다. 그리고 신경망 에이전트를 소개하고 이를 실제 웹 분산 정보 검색 환경에서 구현하여 성능을 평가하고 이를 기존 기법들의 결과와 비교/분석한다.

2. 관련 연구

본 절에서는 분산 정보 검색 시스템에서 데이타베이스 선택 문제를 해결하기 위해 사용하는 대표적인 자동화된 기법들을 기술한다. 이를 위하여 1 절에서 간단히 소개한 시스템들 중에서 통계적 방법들을 사용하는 SMART, *GlOSS* 시스템과 증강 학습법을 사용하는 SavvySearch 에 대해서 보다 자세히 살펴본다. 이러한 시스템들은 모두 데이타베이스 선택 문제를 해결하기 위해 주어진 사용자 질의에 대한 각 문서 데이타베이스의 관련도를 계산하고 충분히 큰 관련도를 가지는 문서 데이타베이스들만을 선택하여 정보 검색을 수행하는 방

1) <http://www.cs.colorado.edu/home/mcbryan/WWWW.html>
2) <http://web.nexor.co.uk/aliweb/doc/aliweb.html>

법을 사용한다. 따라서 본 절에서는 각 시스템들이 주어진 사용자 질의에 대한 각 문서 데이타베이스의 관련도를 어떻게 계산하는지를 중심으로 살펴본다.

2.1 SMART 시스템

SMART 시스템에서는 군집 파일 (clustered file) 이라고 불리는 각 문서 데이타베이스를 그 문서 데이타베이스 내의 모든 문서들에 대한 중심점 용어 벡터로 나타낸다.

이를 위하여 먼저 모든 문서들로부터 의미를 가지는 단어들의 모든 어근(root)들을 용어들로서 추출하여 용어 집합 $T=\{t_1, t_2, \cdots, t_N\}$을 만든다. 이를 이용하여 문서 d는 다음과 같은 조건을 만족하는 용어 벡터 $V_d=(v_{d1}, v_{d2}, \cdots, v_{dN})$ 로 나타낸다.

$$\text{for } i=1,2,\cdots,N, \quad v_{di}=\begin{cases} 1 & \text{if } t_i \text{ occurs in } d \\ 0 & \text{otherwise} \end{cases}$$

문서와 마찬가지로 질의 q 도 다음과 같은 조건을 만족하는 용어 벡터 $V_q=(v_{q1}, v_{q2}, \cdots, v_{qN})$로 나타낸다.

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

문서 데이타베이스 db 에 존재하는 모든 문서들의 집합 D_{db}에 대하여 db의 중심점 용어 벡터 C_{db}를 다음과 같이 구한다.

$$C_{db}=\frac{\sum_{d\in D_{db}} V_d}{|D_{db}|}$$

그러므로 SMART 시스템에서 각 문서 데이타베이스는 그 문서 데이타베이스 내에 존재하는 모든 문서들을 나타내는 용어 벡터들의 중심점 벡터(centroid vector)로 표현된다.

이제 주어진 사용자 질의 q에 대한 문서 데이타베이스 db의 관련도 $R_{q,db}$를 다음과 같이 계산한다.

$$R_{q,db}=\frac{V_q \cdot C_{db}}{\|V_q\| \times \|C_{db}\|}$$

따라서 $R_{q,db}$는 벡터 V_q와 벡터 C_{db}가 이루는 각 θ에 대한 COSINE θ 값을 의미한다. 이러한 방법은 대규모 문서들이 문서 데이타베이스들에 잘 분류되어 각 문서 데이타베이스 내의 문서들이 용어 벡터 공간에서 효과적으로 군집, 분포하는 경우에 상당히 좋은 결과를 낳게 될 것이다. 그러나 이러한 방법은 문서 데이타베이스 내의 문서들이 용어 벡터 공간에서 어떻게 분포하는지를 전혀 고려하지 않고 하나의 중심점 벡터로 문서 데이타베이스를 표현하므로 실제 분산 정보 검색에서는 종종 잘못된 결과를 낳는다. 그러므로 주어진 질의에 대한 문서 데이타베이스의 더 정확한 관련도 계산을 위해서는 그 문서 데이타베이스 내의 문서들에서의 용어들의 분포 형태를 고려해야 할 것이다.

2.2 *GlOSS* 시스템

GlOSS 시스템은 각 문서 데이타베이스에 대하여 다음과 같은 두 가지 통계 정보를 유지하여 주어진 질의에 대하여 반환하는 관련된 문서의 개수를 추정하고 이를 주어진 사용자 질의에 대한 문서 데이타베이스의 관련도로서 사용한다.

DBSize(*db*) : 문서 데이타베이스 *db*에 존재하는 문서들의 전체 개수.

freq(*t*, *db*) : 문서 데이타베이스 *db*에 존재하는 용어 *t* 가 나타나는 문서들의 개수.

일반적으로 각 문서 데이타베이스들은 각 용어에 대하여 그 용어가 나타나는 문서들의 리스트를 색인 정보로서 가지고 있다. 이러한 각 문서 데이타베이스가 가지고 있는 방대한 양의 모든 색인 정보를 유지하는 것은 데이타베이스 선택 문제의 현실적인 해결 방법이 아니므로 *GlOSS* 시스템에서는 각 문서 데이타베이스가 색인 정보로서 가지고 있는 각 용어에 대한 문서 리스트의 크기(즉 각 용어에 대하여 그 용어가 나타나는 문서들의 개수)만을 유지한다. 이와 같은 각 용어에 대한 통계 정보는 각 문서 데이타베이스가 미리 정해진 규약(protocol) 에 따라 *GlOSS* 시스템에 주기적으로 제공하거나 *GlOSS* 시스템이 각 문서 데이타베이스에 주기적으로 단일 용어로 된 질의를 주어서 반환되는 문서들의 개수를 알아내야 한다.

GlOSS 시스템은 각 용어에 대한 위와 같은 통계 정보를 이용하여 각 문서 데이타베이스가 주어진 질의에 대하여 반환하는 관련된 문서의 개수를 추정하기 위해 다음과 같은 가정을 기반으로 한다.

○ *각 문서 데이타베이스는 항상 정확한 색인 정보를 가지고 있다. 따라서 주어진 질의에 대하여 반환하는 문서들은 모두 그 질의에 관련된다.*

○ *각 용어는 문서 데이타베이스 내의 문서들에서 독립적 (independently), 균일적 (uniformly) 으로 분포한다.*

이상적으로 생각한다면 주어진 질의에 관련된 문서는 그 질의를 준 사용자가 흥미를 가지는 문서이다. 그러나 현실적으로 질의를 준 사용자 외에는 이것을 알 수가 없으므로 *GlOSS* 시스템에서는 첫 번째 가정과 같이 주어진 질의에 관련된 문서들을 각 문서 데이타베이스가 그 질의에 대해서 제공하는 모든 문서들로 가정하는 것이다.

이러한 가정 하에서 용어 $t_1, t_2, \cdots, t_n$들로 이루어진

질의 "find $t_1 \wedge t_2 \wedge \cdots \wedge t_n$"에 대하여 문서 데이타베이스 db가 반환하는 관련된 문서의 개수의 추정치 $Esize$ (find $t_1 \wedge t_2 \wedge \cdots \wedge t_n$, db)를 다음과 같이 계산한다.

$$Esize(\ \text{find}\ t_1 \wedge t_2 \wedge \cdots \wedge t_n\ , db\ ) = \frac{\prod_{i=1}^{n} freq(t_i, db)}{DBSize(db)^{n-1}}$$

이러한 $Esize(\ \text{find}\ t_1 \wedge t_2 \wedge \cdots \wedge t_n\ , db\ )$ 값은 사용자 질의 "find $t_1 \wedge t_2 \wedge \cdots \wedge t_n$"에 대한 문서 데이타베이스 db의 관련도로서 사용된다.

위와 같은 각 용어에 대한 통계적 정보를 이용한 *GlOSS* 시스템의 기법은 특정한 조건하에서의 분산 정보 검색에서 널리 사용되고 있으나 이 기법이 기반으로 하는 각 용어의 분포와 관련된 가정은 상당히 제한적이며 때로는 비현실적이다. 왜냐하면 문서 데이타베이스 내의 문서들에서 각 용어의 분포는 많은 경우에 그 문서 데이타베이스의 종류에 따라 달라지기 때문이다. 예를 들어 두 용어 "apple"과 "computer"는 컴퓨터 관련 문서 데이타베이스에서는 주로 같은 문서에서 존재하는 경향이 있으나 일반적인 다른 문서 데이타베이스에서는 그렇지 않다. 결과적으로 문서 데이타베이스 내에서 각 용어가 독립적으로 균등하게 분포한다는 가정은 일반적으로 성립하는 것은 아니다. 그러므로 문서 데이타베이스에 따라 다를 수 있는 각 용어의 분포 상황을 고려하는 데이타베이스 선택 문제의 해결 기법이 요구된다.

2.3 SavvySearch

SavvySearch 는 유명한 여러 웹 검색 엔진들을 하위 문서 데이타베이스들의 색인 시스템으로서 가지는 메타 검색 엔진이다. 또한 어떤 검색 엔진들에 주어진 질의를 보낼 것인가를 경험적으로 학습하는 지능형 메타 검색 엔진이다.

SavvySearch 에서는 모든 용어들의 집합을 T라고 할 때, 각 용어 $t \in T$에 대한 문서 데이타베이스 db의 관련도 $M_{t,db}$(초기값은 0)를 예제 질의 $q \subseteq T$에 대한 정보 검색 결과를 이용하여 다음과 같이 간단한 증강 학습법으로 조정한다.

예제 질의 q에 대해서 문서 데이타베이스 db가 반환하는 관련 문서들의 개수를 $|db(q)|$라 할 때,

$$\forall t \in q \quad M_{t,db} \leftarrow M_{t,db} + \begin{cases} \frac{1}{|q|}, & \text{if } |db(q)| \geq 1, \\ \frac{-1}{|q|}, & \text{if } |db(q)| = 0. \end{cases}$$

이것은 주어진 예제 질의에 대해서 문서 데이타베이스가 관련 문서를 반환하는 경우에는 그 질의를 이루는 각 용어에 대한 문서 데이타베이스의 관련도는 증가하고, 관련 문서를 반환하지 않는 경우에는 그 질의를 이루는 각 용어에 대한 문서 데이타베이스의 관련도는 감소한다는 것을 뜻한다.

충분한 예제 질의들에 대하여 이러한 방법으로 각 용어에 대한 문서 데이타베이스의 관련도를 계속적으로 조정하여 얻어진 결과를 사용하여 새로운 질의 $q \subseteq T$에 대하여 문서 데이타베이스 db의 관련도 $Q_{q,db}$를 다음과 같은 식에 의해 계산한다.

$$Q_{q,db} = \sum_{t \in q} \frac{M_{t,db} \times I_t}{\sqrt{T_{db}}} \quad where \left\{ \begin{array}{c} I_t = \frac{1}{|\{db \mid M_{t,db} > 0\}|} \\ T_{db} = \sum_t |M_{t,db}| \end{array} \right\}$$

그러므로 주어진 질의에 대한 문서 데이타베이스의 관련도는 그 질의를 이루는 각 용어에 대한 문서 데이타베이스의 관련도에 비례하고 그 질의를 이루는 각 용어의 편재도(ubiquity)에는 반비례하며 문서 데이타베이스의 규모의 제곱근에 반비례 한다. 이것은 질의에 대한 문서의 관련도를 계산하는 표준 계산법인 TF×IDF[11]와 같은 방식을 기반으로 하고 있음을 알 수 있다.

이러한 각 용어에 대한 증강 학습을 통한 문서의 관련도를 조정하는 SavvySearch 의 기법은 각 문서 데이타베이스로부터 직접적인 통계 정보를 제공받지 않고도 예제 질의들에 대한 검색 결과들을 통해서 학습을 수행하므로 사용자 질의와 분산된 문서 데이타베이스들의 변화에 능동적 적응성을 가진다.

그러나 예제 질의들을 이루는 용어들의 상호 관련성을 전혀 고려하지 않는 단순 증강 학습법을 사용하므로 상호 간섭 (cross-talk) 을 발생시키는 예제 질의들에 대해서 학습할 경우 바람직한 결과를 낳지 못하게 된다. 예를 들어 4 개의 예제 질의 q_1="information system", q_2="system software", q_3="software tool", q_4="tool information" 에 대하여 문서 데이타베이스 db가 반환하는 관련 문서들의 개수를 각각 $db(q_1) = 7$, $db(q_2) = 0$, $db(q_3) = 7$, $db(q_4) = 0$라고 한다면, 이 4개의 예제 질의에 대하여 SavvySearch 의 증강 학습법에 의해 학습되어진 각 용어 t_1= "information", t_2= "system", t_3= "software", t_4 = "tool" 에 대한 문서 데이타베이스 db 의 관련도는 모두 $M_{t_1,db} = M_{t_2,db} = M_{t_3,db} = M_{t_4,db} = 0$ 이 되어 어떠한 정보도 학습하지 않은 것과 같게 된다. 그러므로 예제 질의들을 이루는 용어들의 상호 관련성을 함께 학습하는 기법이 더 바람직한 결과를 보여 줄 수 있을 것이다.

3. 신경망 에이전트 (Neural Net Agent)

관련 연구에서 살펴본 바와 같이 기존 시스템에는 데

이타베이스 선택 문제를 해결하기 위해 각 용어들에 대한 각 문서 데이타베이스의 관련도를 먼저 계산하고 이를 기반으로 하여 주어진 질의에 대한 각 문서 데이타베이스의 관련도를 추정하는 방법을 사용하며 이러한 방법들은 일상의 데이타베이스 선택 문제에 적용하는데는 각각의 한계점들을 가지고 있다. 이를 극복하기 위해서 본 절에서는 신경망 모델을 적용한 정보 검색 에이전트를 제시한다. 이를 위하여 먼저 인간의 시각 영상 인식 과정에 대한 비유를 통하여 신경망 기법의 필요성에 대하여 언급한다.

3.1 배경

일반적으로 정보 검색에서 용어 기반 검색 방법은 컴퓨터 비전에서 템플릿 기반 비교 방법(template matching method)과 상당한 연관성을 가지고 있다. 인간의 시각적인 인지 과정은 문서를 인식하는 과정과 흡사한 면이 많기 때문에 시각 영상에 대한 인간의 이해 과정을 살펴보는 것은 문서 데이타베이스에 존재하는 문서들이 나타내는 문맥 의미(context semantics)를 이해하는데 도움이 될 것이다.

Biederman은 시각 영상에 존재하는 여러 개의 구성 요소들이 모여서 하나의 문맥(context)을 이루고 어떠한 문맥에도 속하지 않는 구성 요소들은 인간에게 쉽게 인식되지 않음을 보였다[12]. 예를 들어 토우스터(toaster) 기계는 거리 영상(street image)에서는 쉽게 인식되지 않는 반면, 부엌 영상(kitchen image)에서는 중요한 구성 요소로 쉽게 인식된다. 이것은 대략 0.1 초의 짧은 시간 동안 시각 영상을 보여주었을 때의 결과로서, 인간은 영상에 존재하는 각각의 구성 요소들의 이해를 바탕으로 전체적인 영상을 이해하고 또한 전체적인 영상의 이해를 통하여 다시 각각의 구성 요소들을 보다 면밀히 이해하는 과정들을 반복적으로 수행함으로써 전체적인 영상을 정확히 이해한다는 것을 의미한다. 이러한 결과는 시각 영상의 문맥은 그 영상에 존재하는 의미적으로 일관된 구성 요소들의 모임을 뜻한다는 중요한 의미를 담고 있다. 즉 거리 영상에서 거리를 나타내는 어떠한 단일 구성 요소도 존재하지 않으며, 또한 부엌 영상에서도 부엌을 표현하는 어떠한 단일 구성 요소도 존재하지 않는다. 예를 들어 부엌 영상에서 부엌을 표현하는 것은 의미적으로 일관된 토우스터, 냉장고, 싱크대, 가스렌지 등의 모임이지 토우스터 같은 하나의 구성 요소나 의미적으로 일관되지 않는 냉장고, 전봇대, 옷걸이 등의 모임은 아니다.

이와 같은 시각 영상 인식 과정에 대한 분석은 문서 데이타베이스에 존재하는 문서들이 나타내는 문맥 의미를 이해하는데도 적용될 수 있다. 문서 데이타베이스에 존재하는 각 문서의 문맥 의미는 그 문서를 구성하는 의미적으로 일관된 용어들의 모임에 의해 나타내어 진다. 결과적으로 주어진 질의에 대한 문서 데이타베이스의 관련도는 질의를 이루는 용어들이 문서 데이타베이스에 존재하는 여러 문서들의 문맥 의미들을 얼마나 충분히 표현하는가를 평가함으로써 구해져야 한다. 즉 문서 데이타베이스와 관련되는 것은 각각의 용어들이 아니라 그 문서 데이타베이스에 대하여 의미적으로 일관된 용어들의 모임이며, 따라서 주어진 질의에 대한 문서 데이타베이스의 관련도 평가는 질의를 이루는 의미적으로 일관된 용어들의 모임이 문서 데이타베이스와 얼마나 관련되어 있는지를 기반으로 해야 할 것이다. 그러므로 질의를 이루는 각각의 용어에 대한 문서 데이타베이스의 관련도만을 사용하여 질의 관련도를 평가하는 기존의 방법들은 데이타베이스 선택 문제의 일반적인 해결 방법으로 적합하지 못하며 따라서 본 연구에서는 질의를 검색하고자 하는 문서의 문맥 의미를 나타내는 패턴으로서 사용하는 신경망 에이전트를 제시한다.

3.2 구조

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

질의 브로드캐스터(QB: Query Broadcaster): QB는 질의 제공자로부터 주어진 질의에 대하여 관련된 문서

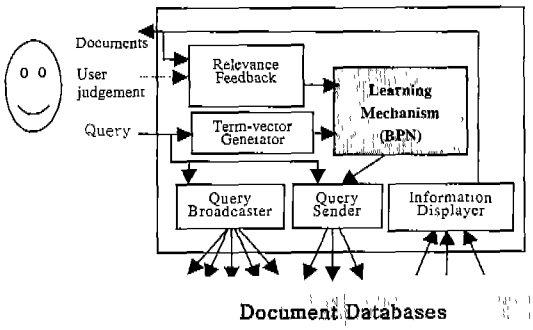

그림 1 신경망 에이전트의 구조

들을 제공받기 위해 존재하는 모든 문서 데이타베이스들에 주어진 질의를 브로드캐스트한다.

정보 디스플레이어 (ID: Information Displayer) : *ID* 는 문서 데이타베이스들로부터 반환되어지는 문서들의 중복성을 제거하여 합병한 결과(반환되어진 모든 문서들의 합집합)를 사용자에게 질의 제공자에게 디스플레이 한다.

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

$$\text{for } i=1,2,\cdots,M,\quad c_{qi}=\begin{cases}\dfrac{m_{qi}}{\max_{j=1,\cdots,M} m_{qj}} & \text{if } \max_{j=1,\cdots,M} m_{qj}\neq 0\\ 0 & \text{otherwise}\end{cases}$$

이 성립한다.

m_{qi}는 질의 q에 대하여 문서 데이타베이스 d_i가 반환하는 문서들의 개수를 구함으로써 쉽게 구할 수 있다. 그러나 문서 데이타베이스의 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 데이타베이스가 반환하는 문서들이 모두 질의 제공자가 원하는 것은 아니라는 것이다. 본 신경망 에이전트는 여러 문서 데이타베이스들로부터 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 데이타베이스의 검색 결과에 대하여 질의 제공자로부터 관련도 피드백을 얻음으로써 구할 수 있다. 본 연구의 후반에서 소개되는 신경망 에이전트의 실험에서는 이와 같이 질의 제공자로부터의 관련도 피드백을 이용하는 방법을 택하고 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[5]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,

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

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. 그림 2는 신경망 에이전트 내부의 BPN이 훈련 단계(training phase)와 검색 단계(retrieval phase)의 두 가지 단계로 동작한다는 것을 보여준다.

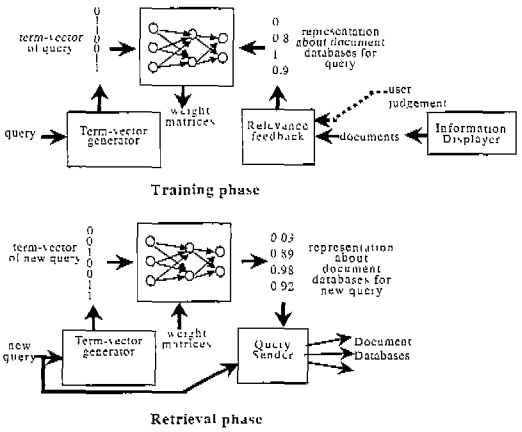

그림 2 BPN의 훈련 단계와 검색 단계

훈련 단계에서 BPN의 입력층은 *TG*에 의해 생성되어진 S_q를 나타내고, 출력층은 *RF*에 의해 생성되어진 C_q를 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[13]을 적용한다. 신경망 에이전트에서 역전파 알고리즘은 BPN 내부 노드들간의 연결 가중치들을 조정함으로써 각 문서 데이타베이스가 제공하는 문서들의 관련도에 대해 학습한다.

검색 단계에서는 주어진 질의에 대하여 *TG*에 의해 생성되어진 용어 벡터가 BPN 의 입력층에 주어지고 이

는 훈련 단계에서 학습되어진 노드들간의 연결 가중치를 기반으로 하여 은닉층을 통하여 출력층으로 전파된다. 이의 결과로 BPN은 0과 1사이의 값들로 이루어지는 벡터를 출력한다.

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

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

3.3 훈련

신경망 에이전트의 훈련 절차는 문서 데이타베이스들로부터 반환되는 문서들에 대하여 관련도 피드백을 받아 주어진 질의에 관련되는 문서가 어디에 있는지에 대한 지식을 습득한다.

*QB*는 주어진 질의를 모든 이용 가능한 문서 데이타베이스들에게 브로드캐스트하고 문서가 반환되기를 기다린다. *ID*는 반환되는 모든 문서들을 합병하고 질의 제공자에게 디스플레이한다.

*RF*는 *ID*에 의해 디스플레이되는 문서들로부터 주어진 질의에 관련된 문서들을 검사하고 각 문서가 어느 문서 데이타베이스로부터 왔는지를 알아내어 결과적으로 각 문서 데이타베이스가 주어진 질의에 관련된 문서들을 얼마나 많이 제공하였는지를 표현하는 벡터를 생성한다. 주어진 질의에 대하여 *RF*에 의해 생성되는 벡터는 *TG*에 의해 생성되는 용어 벡터와 함께 하나의 훈련 쌍을 형성한다.

```
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

이와 같은 방법으로 질의 제공자에 의해 주어지는 모든 훈련 질의들로부터 생성되는 훈련 쌍들의 집합으로 신경망 에이전트 내부의 BPN을 훈련시킨다. 그림 3은 질의 제공자에 의해 주어지는 훈련 질의들에 대한 신경망 에이전트 α의 전체 훈련 절차를 보여준다.

3.4 정보 검색

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 q를 용어 벡터 S_q로 반환한다.

단계 2: S_q에 의해 활성화된 *LM*(BPN)는 검색 단계에 의해 O_q를 출력한다.

단계 3: *QS*는 O_q를 기반으로 하여 문서 데이타베이스들을 선택하고 선택된 문서 데이타베이스들에 q를 보낸다.

단계 4: *ID*는 단계 3에서 선택된 문서 데이타베이스들로부터 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 데이타베이스들을 찾아내고 그 문서 데이타베이스에 질의를 보내어 정보 검색을 수행한다.

4. 실 험

4.1 실험 환경 및 구성

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 데이타베이스로서 동작한다. 그러므로 제안된 신경망 에이전트 기법의 성능을 실제 분산 정보 검색 환경에서 평가하기 위해 야후! 코리아의 검색 디렉토리들을 문서 데이타베이스들로 사용하는 신경망 에이전트 기반 정보 검색 시스템을 구현하였다. 표 1은 본 실험에서 사용된 16개의 문서 데이타베이스(*야후! 코리아*의 디렉토리)들을 보여주고 있다.

위와 같은 16개의 문서 데이타베이스 위에서 작동하

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

는 신경망 에이전트 정보 검색 시스템(SNA : Single Neural net Agent information retrieval system)을 구축하였다. 웹을 통해 접근하여 사용할 수 있게 한 SNA에 26일 동안 133명의 사용자가 용어들의 부울 논리곱(boolean AND)형태의 797 개 질의를 주었다. 각각의 질의에 대하여 SNA는 훈련 단계에서 16개의 문서 데이타베이스들을 무차별적(exhaustive)으로 검색하여 사용자의 관련도 피드백으로부터 몇 개의, 때로는 0개의, 관련된 문서들을 확인하였다. 이 때, 사용자에게 신경망 에이전트의 ID에 의해 제시된 문서들을 모두 검사하도록 요청하였다. 그림 4와 그림 5는 SNA에서 질의를 주고 그에 대해 반환된 문서 요약들에 대하여 사용자로부터 관련도 피드백을 받아들이는 사용자 인터페이스를 보여주고 있다.

이러한 과정을 통하여 사용자가 실제로 신경망 에이전트의 *ID*에 의해 제시된 문서들을 모두 검사하여 적어도 하나 이상의 관련된 문서가 확인된 질의들만을 실험에

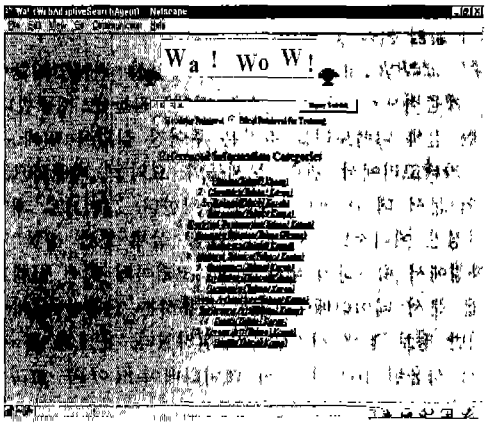

그림 4 신경망 에이전트의 질의 입력 인터페이스

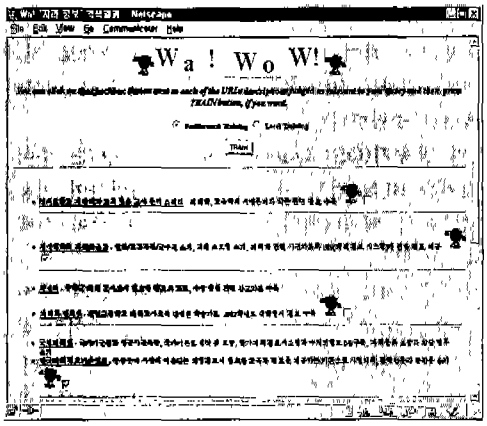

그림 5 신경망 에이전트의 관련도 피드백 인터페이스

서 고려하였다. 결과적으로 질의와 그에 관련된 문서들로 이루어진 734개의 예제 쌍들을 수집하였다. 수집된 예제 쌍들에서 질의들의 토픽은 16개의 문서 데이타베이스 카테고리들을 고르게 포함하였고 각 질의에 대하여 관련된 문서들은 하나의, 때때로 불과 몇 개의, 문서 데이타베이스에 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.

본 실험에서 BPN 입력층의 크기는 734개의 질의들에 나타나는 색인 용어들의 개수[3], 221,로 하였고 BPN 출력층의 크기는 문서 데이타베이스들의 전체 개수, 16,으로 하였다. 다른 BPN 파라미터들의 값은 표 2에서 나타내었다.

훈련 예제들로부터 657개의 훈련 질의들과 그에 대한 사용자의 관련도 피드백 결과를 생성하고 이들을 사용

표 2 BPN 파라미터

Structural Parameters		Operational Parameters	
Input Layer	221	Learning Rate	0.005
Hidden Layer	100	Initial Bias Weight	0.2
Output Layer	16	Maximum Acceptable Average Squared Error	0.05

3) 질의에 나타나는 용어들에 대한 불용어 제거 과정, 복수 명사를 단수로 바꾸거나 변형된 동사를 원형으로 바꾸는 등의 스테밍 과정, 그리고 용어 빈도(term-frequency), 문서 빈도(document-frequency)와 같은 각 용어에 대한 통계 정보를 이용한 전처리(preprocessing) 과정[5]을 통해서 색인 용어들의 개수를 조정하였다.

하여 SNA를 훈련시키고 테스트 예제로부터의 77개의 테스트 질의들에 대하여 각각 성능을 평가하였다.

제시된 신경망 에이전트 기반 정보 검색 기법을 2절에서 소개한 기존의 시스템들과 비교하기 위해서 SNA에서 사용된 것과 같은 문서 데이타베이스들과 테스트 예제들에 대하여 실험적으로 구현한 기존 데이타베이스 선택 기법 (SMART, *GlOSS*, SavvySearch)들의 성능도 평가하였다. 기존 시스템들의 데이타베이스 선택 기법들은 각 질의에 대하여 0 보다 크거나 같은 수로 각 문서 데이타베이스의 순위 값 (rank) 을 생성한다. 이러한 기존 시스템들이 생성하는 순위 값들을 본 연구에서 제시한 방법과 서로 비교할 수 있게 하기 위해 의해 다음과 같이 0과 1 사이의 값으로 정규화하였다.

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

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

이렇게 정규화 되어진 순위 값 n_{qi}을 실제 분산 정보 검색에 적용하기 위해 주어진 질의를 어떠한 0과 1 사이의 허용 오차 상수 τ에 대하여 $n_{qi} \geq \tau$인 문서 데이타베이스 d_i에만 질의를 보내는 전송 메커니즘으로서 3.2절에서 기술한 신경망 에이전트의 *QS*를 사용하였다.

각 기법들을 사용하여 분산 정보 검색을 수행한 결과에 대한 성능을 평가하기 위한 척도로서 정확률(precision)과 재현률(recall)을 다음과 같이 정의하였다.

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

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

본질적으로 어떠한 검색 알고리즘도 주어진 질의에 대하여 어떤 문서 모임으로부터 관련된 모든 문서들을 항상 반환한다는 것을 보장하지는 못한다. 그러므로 본 실험에서 테스트 예제에 존재하는 주어진 질의에 관련된 문서들은 16개의 문서 데이타베이스들의 문서 모임들에 있는 모든 관련된 문서들이 아닐 수도 있다. 그러나 위와 같이 몇 개의 관련된 문서들을 잃어버리는 것은 야후! 코리아에서 사용하는 검색 알고리즘 때문이지 특정한 데이타베이스 선택 기법에서 연유한 것은 아니다. 그러므로 데이타베이스 선택 기법만의 재현 능력 (recall ability)을 평가하기 위해 재현률을 정의하는 식에서 분모는 주어진 질의에 대하여 16개의 문서 데이타베이스에 존재하는 모든 관련된 문서들의 개수가 아니라 모든 문서 데이타베이스들을 무차별적으로 검색했을 때 실제로 반환되는 관련된 문서들의 개수, 즉 테스트 예제에 존재하는 주어진 질의에 관련된 모든 문서들의 개수, 로서 주어진다.

4.2 실험 결과

*QS*의 허용 오차 상수 τ를 다양하게 변화시키면서 77개의 테스트 질의들 전체에 대한 성능을 평가한 결과를 그림 6에서 나타내었다. 이 그림에서 77개의 테스트 질의들 전체에 대하여 얻어진 평균 정확률과 재현률을 각 τ값에 대하여 나타내고 있다.

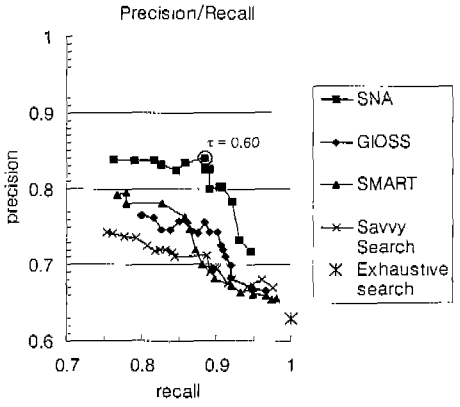

그림 6 정확률과 재현률 평가 결과

일반적으로 예상되는 바와 같이 *QS*의 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 데이타베이스 선택 기법들이 항상 사용자 질의를 모든 문서 데이타베이스들에 무차별적으로 브로드캐스트하게 되며 이 때의 정보 검색 성능은 그림 6에서 무차별 검색 (exhaustive search) 으로 나타내고 있다.

그림으로부터 신경망 에이전트 기반 기법들, SNA이 기존의 기법들, SMART, *GlOSS*, SavvySearch,에 비해 정보 검색 성능이 뛰어나고 한편 *GlOSS*는 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.

τ가 0.60일 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[14]. 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 Workstation* 에서 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.

5. 결론 및 향후 연구

본 연구에서는 분산 정보 검색의 데이터 베이스 선택 문제 해결을 위한 신경망 에이전트 기법을 제시하였다. 제시된 에이전트는 신경망 메커니즘을 정보 검색 지식 습득을 위해 이용하여 사용자 질의에 관련된 문서들을 제공하는 문서 데이타베이스를 찾아내고 그들로부터 정보 검색을 수행한다. 실험으로부터 신경망 에이전트 기법이 기존의 방법들에 비해 현저하게 성능을 향상시킬 수 있음을 확인하였다.

이러한 성능 향상은 신경망 에이전트 기법은 질의에 대한 문서 데이타베이스의 관련도 학습을 위해 기존의 기법들과는 다르게 각 질의를 하나의 패턴으로서 학습함으로써, 질의를 이루는 용어들의 상관 관계가 각 데이타베이스 내에서 상호 배제적인지, 공존적인지, 또는 독립적인지에 대한 정보를 습득할 수 있기 때문에 이루어진다. 실제로 본 실험을 통하여 학습된 신경망 메커니즘은 "단층" 과 "촬영" 이라는 각 용어에 대한 의학 문서 데이타베이스의 관련도는 지리학 문서 데이타베이스에 비해 낮게 평가하였지만 두 용어가 같이 나타나는 "단층 촬영" 이라는 질의에 대한 의학 문서 데이타베이스의 관련도는 지리학 문서 데이타베이스에 비해 매우 높게 평가함으로써, 두 용어 "단층"과 "촬영"은 의학 문서 데이타베이스 내에서는 상호 공존적으로 분포하고 지리학 문서 데이타베이스 내에서는 상호 배제적으로 분포함을 확인할 수 있었다.

본 논문에서 제시한 신경망 에이전트 기법은 기존의 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 데이타베이스를 찾을 수 있다. 이 때, 사용자는 단일 사용자이거나 공통된 흥미를 가지는 사용자 그룹을 뜻하며 신경망 에이전트는 사용자에게 종속적인 정보 검색 에이전트로서 동작한다. 본질적으로 사용자의 흥미는 매우 주관적이므로 주어진 질의에 대하여 문서들이 사용자의 흥미에 관련되는 지에 대한 정확한 결정은 사용자에 의해서만 행해질 수 있다. 따라서 대부분의 기존 기법들은 사용자로부터의 관련도 피드백을 이용하지 않으므로 사용자의 흥미와 관련된 문서들을 제공하는 문서 데이타베이스를 찾는데 효과적이지는 않다.

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

본 논문에서 제시한 신경망 에이전트가 웹과 같은 거대 규모의 분산 정보 검색 환경에서도 효과적인 기법이 되기 위해서는 문서 데이타베이스의 개수의 증가에 대해서 충분히 확장 가능한 특성을 가져야 한다. 예를 들어 이용 가능한 문서 데이타베이스의 개수가 일정 한도를 넘어서게 되면 신경망 에이전트는 BPN을 역전파 알고리즘으로 훈련시키는데 어려움을 겪게 될지도 모른다.

실제로 하위의 문서 데이타베이스들의 개수가 많아질수록 신경망 에이전트의 학습 메커니즘은 각 질의를 관련되는 문서 데이타베이스들에 연관시키기 위해 문서 데이타베이스들 사이를 구별하는 더 많은 특징을 추출해야 하며, 따라서 BPN 훈련을 위한 작업의 규모와 복잡도는 증가한다. Tesauro와 Janssens[15]는 BPN의 훈련 비용(BPN 훈련을 위해 걸리는 시간)이 그 작업의 복잡도에 대해 지수적으로 증가함을 보였다. 또한

Minsky와 Papert[16]는 대부분의 신경망들은 학습해야 하는 대상의 규모를 확장할수록 훈련하는데 있어서 어려움을 겪게 될 것이라고 주장하였다.

이러한 문제점들을 극복하기 위해 분산 정보 검색을 위한 지식이 여러 신경망 에이전트들에 분산된 형태로 학습되고 그 신경망 에이전트들이 상호 협력적으로 정보 검색을 수행하는 다중 신경망 에이전트 시스템을 제시하는 연구가 진행 중에 있으며 초기 실험 결과를 발표한 바 있다[17][18]. 또한 본 논문에서 제시된 다중 신경망 에이전트 정보 검색 시스템을 개방형 정보 검색 시스템으로 발전시키기 위하여 동적인 특성을 갖는 개방형 정보 검색 에이전트 프레임워크와 이에 대한 메커니즘에 대한 연구도 계획하고 있다. 이러한 개방형 프레임워크는 각 신경망 에이전트들이 동적으로 참여하거나 떠날 수 있고 문서 데이타베이스들이 추가, 삭제되거나 문서 데이타베이스들의 내용과 구조들이 비동기적으로 변할 수 있는 환경을 제공한다. 이를 위하여 웹 상에 분산된 문서 데이타베이스들로부터 정보 검색 지식을 자동으로 추출하는 웹 로봇과 이를 적용한 적응형 신경망 에이전트에 관한 연구를 수행할 것이다. 이와 함께 이미 존재하는 기존의 정보 자원이 개방형 프레임워크에서 문서 데이타베이스로서 참여할 수 있게 하기 위하여 랩핑(wrapping)기술에 대한 연구를 병행할 것이다. 랩핑 기술을 적용하여 구현되는 랩퍼(wrapper)는 검색 질의들을 기존의 정보 자원이 이해할 수 있는 질의나 명령으로 번역하고 정보 자원으로부터의 결과를 신경망 에이전트가 이용할 수 있는 형태로 변환하는 기능을 수행하게 될 것이다. 이러한 랩퍼를 이용함으로써 신경망 에이전트는 단순한 문서 모임뿐만 아니라 관계형 데이타베이스와 같은 고전적인 정보 자원들도 정보 검색을 위해 활용할 수 있게 될 것이다.

참 고 문 헌

[1] B. Clifford Neuman, "The Prospero File System: A global file system based on the Virtual System model," Computer Systems, 5(4), 1992.[1]

[2] Michael F. Schwarz, Alan Emtage, Brewster Kahle, and B. Clifford Neuman, "A Comparison of INTERNET resource discovery approaches," Computer Systems, 5(4), 1992.

[3] B. Kahle and A. Medlar, "An information system for corporate users: Wide Area Information Servers," Technical Report TMC199, Thinking Machines Corporation, 1991.

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

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

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

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

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

[9] Yong S. Choi and Suk I. Yoo, "Neural Network Based Web Information Agent," in Proceedings of ACM CIKM'98 Workshop on Web Information and Data Management, November 1998.

[10] Yong S. Choi and Suk I. Yoo, "Neural Net Agent for Discovering Text Databases on the Web," in Proceedings of International Conference on Advances in Databases and Information Systems, September 1999.

[11] Ian H. Witten, Alistair Moffat, and Timothy C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images, Von Nostrand Reinhold, New York, 1994.

[12] I. Biederman, On the Semantics of a Glance at a Scene, Perceptual Organization, Hillsdale, New-Jersey, Lawrence Erlbaum, 1981.

[13] J. A. Freeman and D. M. Skapura, Neural Networks Algorithms, Applications, and programming Techniques, Addison-Wesley, MA, 1992.

[14] B.A. LaMacchia, Internet Fish, PhD thesis, MIT, MA, 1996.

[15] G. Tesauro and R. Janssens, "Scaling relationships in back-propagation learning," Complex Systems, Vol. 6, 1988.

[16] M. Minsky and S. Papert, Perceptrons, MIT Press, Cambridge, MA, 1969.

[17] 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, January 1999.

[18] Yong S. Choi, Suk I. Yoo, and Jaeho Lee, "Hierarchically Organized Neural Net Agents for Distributed Web Information Retrieval," in Proceedings of 23rd International IEEE Computer Software and Applications Conference, October 1999.

최 용 석

1993년 서울대학교 전산과학과(이학사). 1995년 서울대학교 전산과학과(이학석사). 2000년 서울대학교 전산과학과(이학박사). 2000년 3월 ~ 2000년 8월 삼성전자 통신연구소 IMT-2000 연구개발팀. 2000년 9월 ~ 현재 한양대학교 컴퓨터교육과 전임교수. 관심분야는 지능형 정보검색, 소프트웨어 에이전트, 기계학습, 인터넷 응용, 컴퓨터교육