# 코호넨 신경망을 사용한 유즈넷 뉴스 필터링 에이전트 구현†

## (Implementation of Usenet News Filtering Agent using Kohonen Network)

진승훈*, 김종완**, 이승아*, 김영순***,김병만****
(Seung-Hoon Jin, Jong-Wan Kim, Seung-A Lee, Young-Soon Kim, Byeong-Man Kim)

**요 약** 인터넷이 활성화되고 인터넷 사용자도 급증하면서 여러 형태의 많은 정보들이 인터넷을 통해 사용자들에게 제공되어지고 있다. 그 중에서도 많은 뉴스서버들을 통해 제공되는 다양한 뉴스들 중에서 사용자가 원하는 뉴스만 필터링해서 제공받을 수 있는 개인화 서비스에 대한 요구가 증가하고 있다. 본 논문에서는 이러한 뉴스 서비스의 개인화에 대한 요구를 충족시키기 위해 뉴스 필터링 에이전트 시스템을 구현하였다. 구현된 시스템은 코호넨 신경망을 이용해서 사용자가 입력한 키워드에 대해 학습을 실시하여 뉴스그룹을 분류하고, 이를 통해 사용자가 원하는 뉴스만을 제공해 준다. 임의의 사용자를 대상으로 뉴스선호도를 학습한 후 테스트한 결과, 사용자의 선호도를 반영한 뉴스 그룹들을 제시할 수 있었다.

***Abstract*** With the proliferation of internet and an increase in internet users, several kinds of vast information are provided to users on the internet. It is increasing in the need of personalization service by filtering user preferred news among various news documents provided through several news servers. In this paper, we implemented a filtering agent system to meet the demand for personalized news service. In the proposed system, Kohonen network is used to train keywords provided by users and to classify news groups. Resulting from that, the personalized new service is achieved. After we trained and tested the filtering agent, we could provide users news groups with their intention.

## 1. 서 론

1990년대 이후 인터넷이 급속도로 발전하고, 일반 사용자들에게 보급되면서 인터넷을 통해 제공되는 정보의 양도 기하급수적으로 증가하고 있다. 하지만 사용자의 입장에서 보면 아직도 웹 상에서 존재하는 많은 자료들 중에서 찾고자 하는 정확한 정보를 빠른 시간 안에 검색할 수 있다거나, 원하는 정보만 필터링 되어져서 제공받고 있다고 할 수는 없다. 특히 인터넷 사용자들이 많이 사용하는 기능 중의 하나인 뉴스서비스의 경우 매일매일 사용자에게 전달되는 많은 뉴스와 스팸메일을 포함한 광고들 중에서 실제적으로 필요로 하는 뉴스를 검색해 내는 필터링의 기능이 절실히 요구되고 있다.

텍스트로 구성된 문서들 중에서 사용자가 원하는 내용이나 키워드를 포함한 문서만을 필터링해서 제공하는 기능은 이미 오래 전부터 정보검색이라는 분야에서 활발히 연구되어져 왔다[1]. 이러한 정보검색 기능은 이제 인터넷을 통해 제공되는 많은 정보들 중에서 사용자의 요구에 맞는 정확한 문서를 검색해 제공하는 개인화 서비스의 범위까지 확대되고 있으며 이를 위해 지능형 에이전트를 도입하는 연구가 활발히 진행되고 있다[2,3].

개인화 서비스는 인터넷을 통해 제공받을 수 있는 여러 정보형태 중 웹 문서, 메일, 뉴스 등에 대해 이루어질 수 있다. 그 중에서도 수많은 뉴스서버들에서 매일매일 엄청난 양으로 무작위의 사용자들에게 제공되어지는 뉴스에 대한 필터링은 개인화 서비스 중에서도 중요한 분야라고 할 수 있다.

본 논문에서는 수많은 뉴스서버들에서 제공하는

* 대구대학교 대학원 컴퓨터정보공학부
** 대구대학교 정보통신공학부 교수
*** 포항1대학 컴퓨터정보처리과 교수
**** 금오공과대학교 컴퓨터공학부 교수
† 이 논문은 2002학년도 대구대학교 학술연구비 지원에 의한 논문임.

뉴스들 중 사용자가 원하는 정확한 뉴스만을 필터링 해주는 서비스에 대한 사용자 요구를 해결하기 위해 먼저, 인터넷에 접속된 뉴스서버들에 접속해서 뉴스를 모아오도록 한다. 사용자는 미리 관심있는 분야에 대한 키워드를 입력할 수 있고, 시스템이 각 뉴스서버들에서 모아온 뉴스들 중에서 사용자가 입력한 키워드에 맞는 뉴스를 걸러낼 수 있도록 뉴스 필터링 시스템을 구현하였다. 또한 이 시스템에서는 사용자가 입력한 키워드를 통해 사용자의 기호를 학습하여 뉴스를 필터링하기 위해 신경망 기법 중에서 코호넨 신경망을 이용하였다. 코호넨 신경망은 지속적인 사용자의 피드백을 요구하지 않는 비지도 학습의 한 종류로 사용자가 입력한 키워드만를 가지고 뉴스그룹들을 학습시킬 수 있어서, 프로파일을 이용한 뉴스 그룹의 순위를 부여할 수 있다는 장점이 있다. 이에 본 연구에서는 코호넨 신경망을 필터링 알고리즘으로 채택하였다.

본 논문의 구성을 살펴보면 먼저 2장에서 기존의 관련 연구를 통하여 코호넨 신경망과 사용자의 기호를 이용한 뉴스 필터링 시스템에 대해 살펴본다. 3장에서는 본 논문에서 제안한 사용자 기호를 학습해서 뉴스를 필터링 하도록 구현한 시스템을 설명하고, 4장에서는 시스템을 실제 구현하고, 실험결과를 평가한다. 그리고 5장에서는 논문의 결론 및 향후 연구 방향을 제시한다.

## 2. 관련 연구

### 2.1 신경망

인간의 두뇌 작용을 신경세포들간의 연결관계로 모델링한 것을 신경망(neural network)이라 하는데, 신경망은 사람의 학습 능력과 마찬가지로 교사(teacher, trainer)가 가르쳐주면서 학습하는 지도학습(supervised learning) 신경망과 교사없이 스스로 학습하는 비지도 학습(unsupervised learning) 신경망으로 분류할 수 있다[4].

지도 학습은 실험 데이터의 입력벡터와 그에 대응하는 출력값을 함께 신경망에 입력시킨 후 학습시키는 방법이다. 따라서 학습 전에 이미 엄선된 각 단계별 데이터와 입력된 데이터의 출력 결과가 의도한 결과와 일치하는지를 알려주는 교사를 필요로 한다. 항상 학습할 때는 교사와 잘 분류된 학습 데이터가 필요하므로, 데이터의 선정시간이 지나치게 길어지거나 교사가 일일이 학습에 관련해야 하는 등의 단점이 있다. 대표적인 알고리즘들로는 델타규칙(delta rule)과 오류역전파(error backpropagation) 학습 규칙이 있다.

비지도 학습 신경망은 출력층의 목표값을 필요로 하지 않으므로 미리 결정된 해(解)와의 비교가 필요하지 않다. 대신 학습 알고리즘은 비슷한 입력 패턴들이 같은 출력뉴런으로 학습되도록 연결가중치들을 수정하여 준다. 따라서 학습과정은 학습 데이터의 통계적 성질을 추출하고, 유사한 벡터들을 같은 클래스로 분류하여 준다. 이 방식은 단지 주어진 클래스로부터 벡터가 입력되면 특정한 출력벡터를 생성하지만, 어떤 출력 패턴이 주어진 입력벡터에 의해 생성되는지 알 수가 없다. 하지만 이 점은 신경망에 의해 확립된 입출력 관계를 식별하면 되는 단순한 문제이다. 대표적인 알고리즘들로는 코호넨으로 대표되는 경쟁학습(competitive learning) 알고리즘과 Grossberg로 대표되는 ART(Adaptive Resonance Theory) 모델이 있다.

### 2.2 코호넨 신경망

코호넨 신경망에서 학습방법은 먼저 각 뉴런이 연결강도(weight) 벡터와 입력벡터의 거리가 얼마나 가까운가를 계산한다[5]. 그리고 각 뉴런들은 학습할 수 있는 특권을 부여받으려고 서로 경쟁하게 되는데 거리가 가장 가까운 뉴런이 승리하게 된다. 이 승자뉴런의 연결강도 벡터는 입력벡터에 가장 가까운 것으로 이 뉴런만이 출력신호를 보낼 수 있는 유일한 뉴런이 되고, 이 뉴런과 인접한 이웃 뉴런들만이 제시된 입력벡터에 대한 학습이 허용된다.

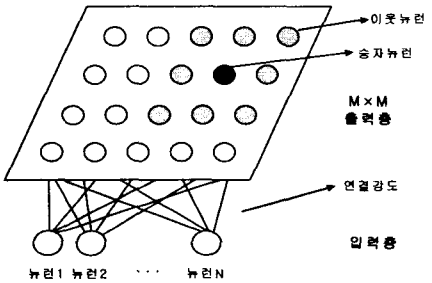


그림 1. 코호넨 신경망

코호넨 신경망의 학습원칙은 단순 경쟁학습이 지향하는 '승자 독점(winner take all)'의 문제점인 일부 뉴런들이 학습되지 않는 현상을 개선하기 위해, 승자 뉴런과 이웃한 뉴런의 연결강도를 함께 학습함으로써 학습되지 않는 뉴런이 생기는 문제를 해결하였다[4].

본 논문에서는 사용자의 기호를 학습하여 원하는 정보를 검색결과로 제시하기 위하여 신경망 중 목표값 없이 학습 데이터만을 단순히 신경망의 입

력으로 사용하여, 신경망이 스스로 연결가중치들을 학습시키는 비지도 학습 회로망의 한 종류인 코호넨 신경망을 사용하였다. 코호넨 신경망을 선택한 이유는 사용자들의 기호를 프로파일로 저장한 뒤, 프로파일만으로 뉴스그룹들을 출력 뉴런의 경쟁층에 자율적으로 배치시키는데 이 방법이 적합하기 때문이다.

### 2.3 필터링 에이전트

정보시스템이 발전하고 사용자에게 제공되는 정보의 양이 증가하면서 사용자가 필요로 하는 정확한 정보를 빨리 검색하여 제공할 수 있는 시스템에 대한 요구가 발생하게 되었고, 정보검색과 필터링(filtering) 시스템 등의 형태로 발전하게 되었다. 또한 이러한 시스템은 인터넷의 사용이 확산되고, 사용자의 시스템에 대한 의존도가 높아지면서 사용자의 요구를 파악하여 그 요구에 대한 작업을 사용자 대신 수행할 수 있는 에이전트(agent) 시스템으로 발전하였다.

현재 인터넷 상에서 운영되고 있는 필터링과 관련된 에이전트는 여러 가지 형태가 있으며, 필터링하는 정보의 종류에 따라 웹 문서 필터링 에이전트, 상용뉴스 필터링에이전트, Usenet 뉴스 필터링 에이전트로 구분한다[6]. 웹 문서 필터링 에이전트는 웹에서 제공되는 문서들 중 특정 분야에 대한 선호도를 기억하고 새로 검색되는 문서들을 분류하여 제공한다. 대표적인 에이전트로 Smart Marks, Point Subscription, WebFilter, WebWatcher, WAIR 등이 있다. WebWatcher[7]는 서버에서 proxy와 같이 실행되며, WAIR[8]는 웹 기반의 개인화된 정보 필터링을 위한 플랫폼이다.

상용뉴스 필터링 에이전트는 인터넷을 통해 제공되는 상업용 뉴스를 사용자에게 제공하는 시스템으로 사용자가 미리 입력한 프로파일에 따라 그에 적합한 새로 추가되는 뉴스를 필터링하여 제공한다. 대표적으로 NewsHound, Personal Journal, PointCast Network[9] 등이 있다.

상용뉴스 서비스와 구분되는 Usenet 뉴스 서비스는 주제별로 구성되며 수많은 사용자들이 자유로이 내용을 게시하고, 무료로 뉴스 서비스를 제공받을 수 있다. 그러나 상용뉴스 서비스에 비해 사용자는 원하지 않는 뉴스까지 모두 제공받게 되므로 Usenet 뉴스 서비스에서의 필터링 기능이 더욱 필요하다고 할 수 있다. 이러한 Usenet 뉴스 서비스에서 사용자의 기호에 맞는 뉴스를 필터링 해주는 에이전트로는 SIFT, NewsClip, MAXIMS, Mailagent 등이 있다. SIFT는 스탠포드 대학에서 개발된 정보 필터링 도구로 사용자의 프로필에 포함된 단어들이 각 문서에 몇 번 나타나는 가를 계산하여 사용자와 문서와의 관련성을 계산한다[10]. NewsClip은 clarinet에서 제공하는 뉴스 필터링 시스템으로 유닉스 프로그래밍 언어로 디자인되었으며 각 뉴스는 순위로 연관성을 표시한다[11]. MAXIMS[12]는 MIT 미디어랩에서 개발된 Apple용 전자우편 필터링 시스템이고, Mailagent[13]는 규칙 기반의 전자우편 필터링 시스템이다.

## 3. 시스템 구조

### 3.1 시스템의 기본 구조

본 논문에서 구현한 뉴스 필터링 시스템은 자바언어로 구현된 Bigus의 뉴스 필터링 시스템[14]을 참고하여 자바언어로 구현하였으며, 사용자 인터페이스를 GUI로 하기 위해 Swing을 사용하였다. 또한 java.net.Socket class를 사용해서 NNTP Server에 접속하였고, NNTP Protocol을 통해서 뉴스그룹을 선택하고, 뉴스문서의 목록 및 내용을 조회할 수 있도록 하였다. 유즈넷 접속과 뉴스그룹, 뉴스문서 조회에 대한 기능을 NewsHost class에 구현하였는데 유즈넷은 news.kornet.net 같은 도메인으로 접속할 수 있는 서버가 있고, 각 서버마다 여러 개의 그룹이 있다. 그러나 존재하지 않는 뉴스그룹이 상당히 많기 때문에 이 프로그램을 사용하여 뉴스서버에서 각 뉴스그룹에 접속할 경우 그 존재하는 뉴스그룹의 경우에는 서버 응답 메시지의 첫 시작이 "211"로 시작한다. 이것을 이용하여 뉴스그룹의 존재 유무를 판단한다.

뉴스서버에서 뉴스를 읽어올 때 먼저 뉴스의 시작번호와 끝번호를 읽어온 후 그 시작번호부터 끝번호까지 뉴스를 읽어오도록 명령어를 실행한다. 이때 처음에 읽어왔던 시작번호와 끝번호의 정보와는 달리 뉴스가 그만큼 존재하지 않는 경우가 종종 있다. 존재하는 문서일 경우 서버 응답 메시지의 첫 부분이 "223"으로 시작한다. 뉴스그룹의 존재 유무의 판단과 같은 방법으로 문서의 유무도 판단한다.

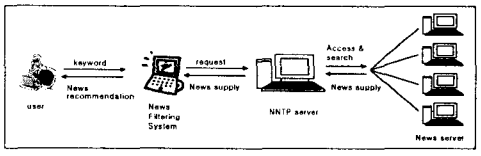


그림 2. 뉴스 필터링 시스템 구조

### 3.2 학습 방법

뉴스 필터링 시스템 신경망 기법 중 코호넨 신경망을 이용하여 사용자의 기호를 학습하게 하였다. 먼저, 사용자는 자신이 원하는 뉴스에 포함될 키워드를 입력할 수 있고, 시스템은 각 뉴스문서에 대해서 각 키워드들이 몇 번 나타나는지를 코호넨 신경망에 대한 입력 백터로 취급해서 학습한다.
코호넨 신경망 학습 알고리즘은 아래와 같이 6단계로 구성된다[5].

[단계 1] 연결강도백터 W를 초기화한다. N개의 입력으로부터 M개의 출력 뉴런 사이의 연결강도를 작은값의 임의의 수로 초기화한다. 이웃반경은 충분히 크게 잡은 후 점차 줄어든다.

[단계 2] 새로운 입력백터 X를 제시한다.

[단계 3] 입력벡터와 모든 뉴런들 간의 거리를 계산한다. 입력과 출력 뉴런 j 사이의 거리 dj는 다 음과 같이 계산한다.

$$dj = \sum_{i=0}^{N-1} [Xi(t) - Wij(t)]^2 \quad (1)$$

[단계 4] 최소거리에 있는 출력 뉴런을 승자 뉴런으로 선택한다. 최소거리 dj 인 출력뉴런 j*를 선택한다.

$$j^* = \min_{j} dj, \quad j \in \text{출력뉴런} \quad (2)$$

[단계 5] 승자 뉴런 j*와 그 이웃들의 연결강도를 재조정한다. 뉴런 j*와 그 이웃 반경내의 뉴런들의 연결강도를 다음 식에 의해 재조정한다.

$$Wij(t+1) = Wij(t) + \alpha \cdot ( Xi(t) - Wij(t) ) \quad (3)$$

$$\alpha = \alpha_0 \cdot (1/epoch) \quad (4)$$

여기에서 j는 j*와 j*의 이웃반경내의 뉴런이고, i는 0에서 N- 1까지의 정수값이다. α는 0과 1사이의 값을 가지는 이득항(gain term)인데 시간이 경과함에 따라 점차 작아진다. 본 연구에서 α값은 초기값 $\alpha_0$로 0.9를 사용하였다.

[단계 6] 단계 2로 가서 반복한다.

## 4. 실험 및 평가

먼저, 훈련 데이터(training data)를 모으기 위하여 자바의 Socket Class를 이용하여 NNTP Server (news.kornet.net)에 접속한 후, 각 뉴스그룹에서 뉴스문서를 내려 받았다. 이때 이미 삭제되었거나 옮겨진 뉴스그룹과 10개 이하의 문서를 가지고 있는 뉴스그룹은 제외시켰다.

실험 결과 131개의 뉴스그룹을 검색하여 조건에 맞는 71개의 뉴스그룹을 훈련데이터로 사용하였으며, 출력뉴런의 크기는 5*5이고, 훈련은 1000회 실시하였다. 훈련 데이터는 각 뉴스그룹에서 임의로 단어들을 선택하여 데이터베이스에 저장해 놓고, 각 뉴스 그룹의 문서를 파싱하여 입력된 단어들의 개수를 알아낸다. 본 논문에서는 총 31개의 단어를 임의로 선출하여 사용하였으며, 그림 3의 경우 각 뉴스그룹의 문서의 수와는 상관없이 단어의 개수만을 파악한 것이므로 문서의 수가 많은 뉴스그룹에서는 대체적으로 단어의 빈도수가 많다. 예를 들어, "han.comp.os.linux.networking" 뉴스그룹의 경우 문서의 수가 1448개인 반면, "han.answers" 뉴스그룹은 24개의 문서만 데이터베이스에 저장되어 있다. 이런 편차를 줄이기 위하여 본 논문에서는 정규화(normalization)를 수행한다.

정규화는 (각 단어의 개수)/(뉴스그룹에서 각 단어들이 나타난 총합)으로 계산하여 각 단어들이 뉴스그룹 내에서 나타나는 비율로 한다.

예를 들어, "han.answers"에서 각 단어들이 나타난 총합이 416번이며, "메일"이란 단어는 284번 나타났다. 이 경우에 "han.answers"에서 "메일"이라는 키워드의 비율은 "284/416 = 0.682" 가 된다. 나머지 단어들도 마찬가지 방식으로 계산한 결과가 그림 4와 같다.

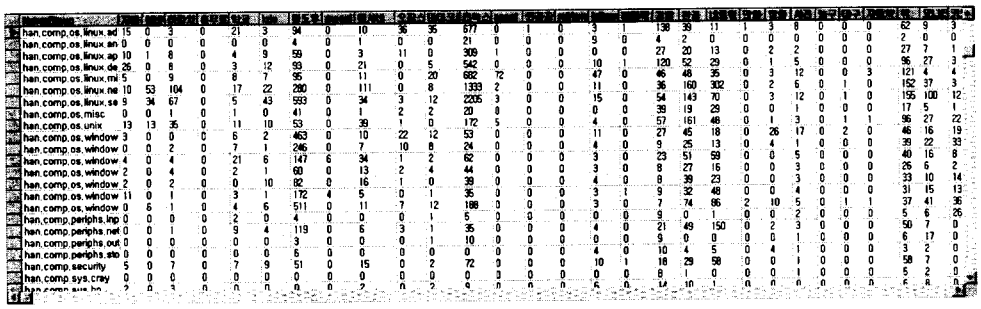

그림 3. 각 뉴스 그룹에 존재하는 입력된 단어의 수

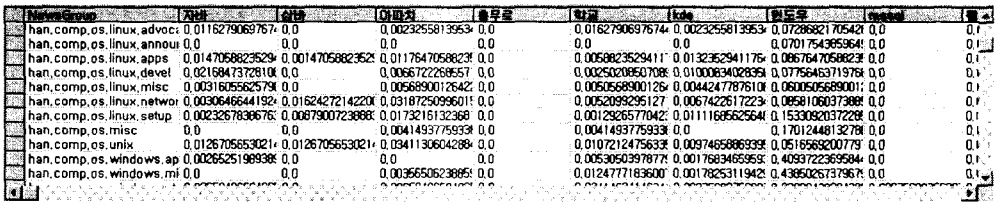


그림 4. 정규화된 입력벡터

그림 4는 본 시스템에서 접속된 각 뉴스그룹 별로 불러온 각 뉴스문서에서 출현한 키워드의 비율을 정규화한 결과를 보여주고 있다. 그림 4의 데이터를 훈련 데이터로 사용하여 신경망을 학습시켰다. 학습은 1000회 동안 반복 수행하였다.

학습 시에 승자 뉴런의 이웃 뉴런의 반경을 radius로 주고, 100회를 기준으로 radius-1 씩 감소시켰으며, 0이 되는 시점, 즉 출력 뉴런 자신만이 훈련되는 시점이 학습의 반복회수(epoch)가 500이 되는 시점이다. 그 이후부터는 오차(error)값이 크게 변하지 않고 있다. 훈련횟수에 따른 제곱 오차합

(E)은 아래의 식으로 정의하였다.

$$Ej = \sum_{i=1}^{D} (W_{ij} - X_{ij})^2 \quad (5)$$

$$E = \sum_{j=1}^{P} Ej \quad (6)$$

여기서 D는 입력벡터의 차원수(실험에서 31), P는 훈련데이터로 사용된 뉴스그룹의 개수(실험에서 71)를 의미한다. Wij는 입력벡터 i와 출력뉴런 j의 연결강도, Xij는 학습데이터, Ej는 j번째 학습데이터의 오차를 나타낸다.

학습의 이득항 a는 식 (4)를 이용하여 초기값 0.9에서 학습회수가 증가할수록 줄어들도록 함으로써, 학습 초기에 값의 변화가 많이 일어나고 있음을 알 수 있다. 즉 이웃 뉴런의 반경(radius)이 변하는 100의 배수 주기 기준으로 오차가 순간적으로 늘었다가 학습이 진행됨에 따라 오차가 줄어들고, 반경이 0이 되는 500회 이후부터는 거의 수렴한다는 사실을 그림 5를 통하여 확인할 수 있다.

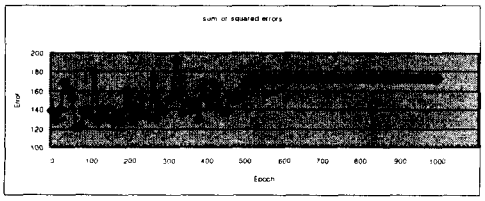


그림 5. 훈련횟수에 따른 제곱 오차합

학습이 끝난 후 각 뉴스그룹의 코호넨 신경망의 출력층 위치와 연결강도 값을 그림 6과 같이 데이터베이스에 저장한다. 그림 6은 학습에 사용된 뉴스그룹들이 학습이 완료된 후 2차원 출력층에 배열된 예의 일부를 보여준다. 그림에서 알 수 있듯이, "리눅스"와 관련된 뉴스그룹들이 코호넨 신경망의 (0,4) 출력 뉴런에 모여 있음을 확인할 수 있다.

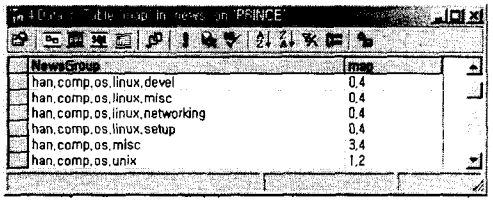


그림 6. 각 뉴스 그룹의 위치정보

그림 7은 사용자가 입력한 키워드 프로파일을 나타낸 것이다. 사용자가 입력한 키워드를 이용하여 테스트용 입력벡터를 생성한다. 사용자가 입력한 키워드와 미리 입력되어있는 키워드와의 거리를 계산하기 위하여 사용자가 입력하지 않은 키워드의 값을 0으로 하여 차원을 일치시켰다. 사용자가 입력한 키워드는 각 뉴스그룹에서 출현한 비율의 평균값을 식 (7)을 이용하여 계산하였다.

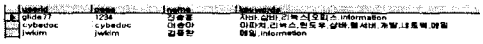

그림 7. 사용자 입력 키워드 프로파일

$$\frac{1}{n} \sum_{i=1}^{n} f_i(k) \qquad \text{for } all \ \ k \quad (7)$$

여기서 k는 키워드 프로파일에 있는 단어, i는 단어 k가 나타난 뉴스그룹, n은 단어 k가 나타난 뉴스그룹의 수를 나타낸다. 따라서 fi(k)는 키워드 k가 i번째 뉴스그룹에 나타난 빈도수 비율을 의미한다.

예를 들면, 표 1에서 사용자 (glide77)가 입력한 단어 중에 "자바"의 경우는 각 뉴스그룹의 비율을 모두 더하여 평균값을 입력 백터로 사용한다. 이때 "자바"라는 단어가 나타나지 않은 뉴스그룹은 제외한다. 식 (7)에 의하여 k="자바"를 계산하면

$f(\text{자바}) = \frac{1}{39}(0.0290 + 0.4904 + \cdots + 0.0056)$

이 되고 결과값은 "0.0833"이 된다.

"아파치"라는 단어의 경우 사용자가 입력하지 않았으므로 "0"으로 둔다. 나머지 단어에 대해서도 같은 방식으로 계산한 결과가 표 1에 나와 있다.

표 1. 사용자 glide77의 테스트용 입력벡터

| | 자바 | 삼바 | 아파치 | 총무로 | 학교 | kde | .. |
|---|---|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | 0 | 0 | [illegible] | 0 | .. |
| han.answers | 0 | 0 | 0 | 0 | 0.03605 | 0 | .. |
| han.arts.architecture | 0 | 0 | 0 | 0 | 0.0666 | 0.02222 | .. |
| han.arts.design | 0.0290 | 0 | 0.00322 | 0.00645 | 0.05483 | 0.00967 | .. |
| han.comp.lang.java | 0.4904 | 0 | 0.01673 | 0 | 0.01711 | 0 | .. |
| ... | ... | ... | ... | ... | ... | ... | .. |
| han.net.kren | 0.0056 | 0 | 0 | 0 | 0 | 0 | .. |

표1의 테스트용 입력벡터가 계산되면 코호넨 신경망에 제시하여 가장 가까운 출력뉴런을 선정하고, 이 뉴런에 속하는 뉴스그룹들을 사용자에게 제시한다.

그림 8은 사용자(glide77)가 자신의 ID를 입력한 후의 결과 화면으로, 사용자가 입력한 키워드와 미리 학습된 정보를 이용하여 가장 가까운 뉴스그룹을 보여준다. 그림 8에서는 출력뉴런 (4,2)가 승자뉴런으로 선정되었다. 본 논문에서는 선정된 승자뉴런과 관련된 뉴스그룹들의 순위(ranking)를 부여하여 사용자에게 순위 순으로 제시한다. 순위는 식 (8)과 같이 (모든 키워드들의 빈도수 비율의 합 / 키워드가 포함된 수)를 이용하여 계산하였다.

$$ranking(k) = \frac{\sum_{i=1}^{p} f_i(k)}{d} \quad \text{for } all\ k \quad (8)$$

p는 프로파일에 등록된 키워드 수, k는 키워드, d는 키워드가 해당 뉴스그룹에 나타난 경우의 수를 나타낸다.

예를 들어, han.comp.lang.java의 경우 ((0.4904 + 0 + 0.0300 + 0 + 0.0015) / 3) = 0.17401이 된다. 표 2의 값은 정규화 되어 그림 4와 같이 데이터베이스에 저장되어 있던 값 중에서 glide77 사용자와 관련된 부분만 보인 것이다. glide77 사용자의 경우 표 3과 같이 순위가 계산되어 그림 8과 같은 결과를 제시한다.

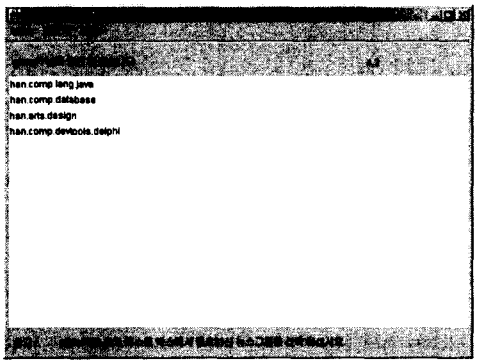


그림 8. 결과 화면 (glide77 사용자)

표 2. 선택된 뉴스 그룹의 순위계산을 위한 값

| | 자바 | 삼바 | 리눅스 | 오피스 | information |
|---|---|---|---|---|---|
| han.comp.lang.java | 0.4904 | 0 | 0.0300 | 0 | 0.0015 |
| han.comp.database | 0.0191 | 0 | 0.1182 | 0 | 0.0086 |
| han.arts.music.gugak | 0 | 0 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... |

표 3. 순위 계산 결과값

| | |
|---|---|
| han.comp.lang.java | 0.17401 |
| han.comp.database | 0.04869 |
| han.arts.design | 0.02580 |
| han.comp.devtools.delphi | 0.018390 |
| han.comp.devtools.vb | 0.00868055555 |
| han.arts.music.gugak | 1.0E-99 |

표 3에서 "han.arts.music.gugak"의 경우 "1.0E-99"의 아주 작은 값이 계산되었다. 코호넨 신경망에서 훈련을 통하여 "han.arts.music.gugak" 뉴스그룹이 표 3에 있는 뉴스그룹과 같은 뉴런에 분류되었으나 사용자가 입력한 키워드가 "han.arts.music.gugak" 뉴스그룹에서 하나도 존재하지 않았으므로 최하위 값이 선정되었다. 본 연구에서는 임계치를 사용하여 임계치 보다 낮은 뉴스그룹은 관련정도가 적다고 판단하여 제외하였다. 현재는 임계치로 0.01을 사용하여, 순위 계산값이 작은 2개의 뉴스그룹을 제거시켰다. 최종 순위 부여 결과가 그림 8과 같이 계산되어 사용자에게 보여진다.

구현된 뉴스필터링 시스템은 기존의 NewsClip 등의 시스템과는 달리 단순히 뉴스들을 필터링하는 것이 아니라 사용자의 선호도에 따라서 뉴스그룹들의 순위를 계산하고 이를 기반으로 추천한다. 따라서 사용자가 자신이 선호하는 뉴스그룹만을 선택하여 정보를 획득할 수 있는 특징이 있다.

## 5. 결론 및 향후 연구 방향

본 연구에서는 사용자가 관심 있는 키워드와 관련 있는 뉴스 그룹을 사용자에게 추천하는 방식으로 유즈넷 뉴스 필터링 시스템을 구현하였다. 학습 방법으로 임의로 선정된 키워드들의 클러스터링에 용이한 코호넨 신경망을 사용하였다.

본 연구의 특징을 다음과 같이 정리할 수 있다. 첫째, 각 뉴스그룹들의 문서의 개수가 서로 달라 비슷한 내용을 지닌 뉴스그룹의 경우라도 문서의 개수가 많은 곳과 적은 곳의 경우 서로간의 단어 빈도수 차이가 많이 나서 거리가 멀어지게되어 비슷한 뉴스그룹으로 분류할 수 없게 된다. 이러한 편차를 줄이기 위하여 정규화를 하였다. 둘째, 테스트 시에 입력벡터의 차원을 일치시키기 위하여, 사용자가 입력한 키워드의 경우, 키워드가 나타난 뉴스그룹들의 빈도수의 평균을 구하여 사용하였다. 셋째, 선택된 뉴스그룹을 사용자에게 순위 순으로 제시하여 어떠한 뉴스그룹이 사용자가 입력한 키워드와 가장 유사한 값을 지니고 있는지를 파악할 수 있으며 순위를 부여하지 않고 제시하는 것보다 불필요한 검색을 줄일 수 있다. 넷째, 비슷한 뉴스그룹으로 분류는 되었으나 사용자가 입력한 키워드와 관계가 적은 뉴스그룹들은 사용자에게 제시할 필요가 없으므로 임계치를 사용하여 제거하였다.

현재의 시스템을 이용하면 사용자는 자신이 관심 있는 분야의 입력한 키워드에 적합한 뉴스그룹들만 볼 수 있게 되므로 불필요한 검색을 줄일 수 있다. 이와 비슷하게 각 뉴스 그룹의 뉴스 문서를 학습 한 후 새롭게 갱신되는 뉴스 문서를 새로운 입력벡터로 사용하여 사용자에게 적당한 문서인지를 파악하여 제공하는 시스템을 추가할 필요가 있다. 또한 본 시스템에서는 테스트를 위하여 입력벡터 키워드를 임의로 선정하였는데, 키워드 추출 방법에 대한 연구도 추가되어야 한다. 또한 사용자가 관심 있다고 등록한 키워드에 대해 새로이 갱신된 뉴스그룹 및 뉴스에 대하여 필터링 한 후, 사용자가 등록한 e-mail 등으로 푸시 서비스하거나, 시스템에 로그인되었을 때 자동으로 서비스할 수 있도록 하는 기능도 추가시킬 필요가 있다.

## 참 고 문 헌


[1] 이승원, 류제, 우성규, 한광록, "지능형 통합 에이전트의 구현," 2000년 한국정보처리학회 추계 학술발표논문집, 제7권 제2호, pp.1437-1440, 2000.

[2] 한선미, 우진운, "지능형 에이전트를 이용한 개인화된 유·무선 뉴스 검색 시스템," 정보처리학회논문지B, 제8-권 제6호, pp.609-618, 2001.

[3] 김태훈, 최중민, "사용자 편의의 인터넷 정보검색을 위한 지능형 웹 브라우징 에이전트," 정보과학회논문지, 제25권 제7호, pp.1064-1078, 1998.

[4] Jongwan Kim, Jesung Ahn, Chong Sang Kim, Heeyeung Hwang, and Seongwon Cho, "Competitive Learning Neural Network with Dynamic Output Neuron Generation," Neural, Parallel & Scientific Computations, Vol.2, No.4, pp.431-450, 1994.

[5] 김대수, 신경망 이론과 응용, 하이테크 정보, 1992.

[6] 최중민, "인터넷 정보가공을 위한 에이전트 연구동향," 정보처리학회지, 4권 5호, pp 101-109, 1997.

[7] T. Joachims, D. Freitag, T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web," Proceedings of IJCAI97, 1997.

[8] Seo, Y.-W., and B.-T.Zhang. 2000. A reinforcement learning agent for personalozed information filtering, In Proc. Int. Conf. on Intelligent User Interfaces(IUI-2000, pp.248-251, New York: ACM Press.

[9] Point CAst Network, http://www.pointcast.com/.

[10] Tak W. Yan, Hector Garcia Molina, "SIFT - A Tool for Wide-Area Information Dissemination," Proceedings of the 1995 USENIX Technical Conference, pp.177-186, 1995.

[11] NewsClip, http://www.clarunet.com/newsclip.html.

[12] Richard B. Segal, Jeffrey O. Kephart, "MailCat: An Intelligent Assistant for Organizing E-Mail," International Conference on Autonomous Agents, 1999.

[13] Mailagent, http://www.nextword.com/MailAgentprod.htm.

[14] Joseph P. Bigus, Jennifer Bigus, Costructing intelligent agents with JAVA, Wiley, 1998.

**진 승 훈** (Seung-Hoon Jin)

email : glide77@hitel.net
2001년 대구대학교 컴퓨터정보공학부 졸업 (학사)
2001년~현재 대구대학교 대학원 컴퓨터정보공학과 재학(석사과정)
관심분야 : 인공지능, 정보검색, XML, 전자상거래

**김 종 완** (Jong-Wan Kim)

email : jwkim@daegu.ac.kr
1987년 서울대학교 컴퓨터공학과 졸업(학사)
1989년 서울대학교 대학원 컴퓨터공학과 졸업(공학석사)
1994년 서울대학교 대학원 컴퓨터공학과 졸업(공학박사)
1995년~현재 대구대학교 컴퓨터정보공학부 부교수
1999년~2000년 미국 U. of Massachusetts Post Doc.
관심분야 : 지능형 에이전트, 퍼지시스템, 인공지능, 전자상거래, 정보검색

**이 승 아** (Seung-A Lee)

email : cybedoc@empal.com
1994년 대구효성가톨릭대학교 경영정보학과 졸업(학사)
1996년 대구효성가톨릭대학교 대학원 경영학과 졸업(경영학석사)
1998년~현재 대구대학교 대학원 컴퓨터정보공학과 박사과정
관심분야 : 전자상거래, 에이전트, 학습

**김 영 순** (Young-Soon Kim)

email : youngsn@pohang.ac.kr
1995년 대구효성가톨릭대학교 경영정보학과 졸업(학사)
1998년 대구대학교 대학원 컴퓨터정보공학과 졸업(공학석사)
1998년~현재 대구대학교 대학원 컴퓨터정보공학과 박사과정
1998년~현재 포항1대학 전산정보처리과 전임강사
관심분야 : 지능정보시스템, 인공지능, 전자상거래, 에이전트

**김 병 만** (Byeong-Man Kim)

email : bmkim@se.kumoh.ac.kr
1987년 서울대학교 컴퓨터공학과 학사
1989년 한국과학기술원 전산학과 공학석사
1992년 한국과학기술원 전산학과 컴퓨터 공학박사
1992년 - 현재 금오공과대학교 부교수
1998년 - 1999년 미국 Univ. of California, Irvine Post Doc.
관심분야 : 인공지능, 정보검색, 소프트웨어 검증